2012

Sensing strategies in cognitive radio networks: optimization and ramifications on routing

Ramzi Rajeh Ahmad Saifan
Iowa State University

Sensing strategies in cognitive radio networks: Optimization and ramifications on routing

by

Ramzi Saifan

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Ahmed E. Kamal, Co-major Professor
Yong Guan, Co-major Professor
Doug W. Jacobson
Sang Kim
Lizhi Wang

Iowa State University

Ames, Iowa

2012

I lovingly dedicate this thesis to my parents who encouraged me day by day during all of my study and helped me to successfully finish the PhD, to my wife for her patience, understanding, and granting me the strength during my critical weak moments, and to my sweetheart kids.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

All praise is due to Allah for his countless blessings during my years of living, for bestowing this Ph.D. degree, and for giving me the hands and the brain to write my dissertation. This dissertation would not have been possible without the guidance and the help of several individuals.

First and foremost, I would like to express my deepest gratitude to my advisors: Dr. Ahmed E. Kamal and Dr. Yong Guan for their guidance, patience and support throughout this research and the writing of this thesis. Their insights and attention to details have always inspired me, while their words of encouragement kept me on the path of completing the graduate education. My special thanks to them for all that they have done for me over the years. Also, I would like to thank my Ph.D. committee members, Professors Doug Jackbson, Lizhi Wang, Lei Ying, and Sang Kim. They provided me with helpful comments and feedback on my research.

I will always be indebted to my parents (Rajeh Saifan and Seham Saifan), who always thinking about me, tremendously supporting me, wishing me all the best, and following my progress day by day overseas. I would like to extend my deepest gratitude to my beloved wife (Raghda Hasan) for her patience and understanding. Raghda has been a rock by my side and has always motivated me to move along at difficult times, she sacrificed all of her time to take care of our daughters (Leen and Hala). Really, without her support, this thesis would not have seen the light. I will forever be grateful to my father- and mother-in-law (Theyab Hasan and Baheya Al-Laham) for their love and the nice wishes they introduced to me. My brothers (Ahmad, Nizar, and Monthir), and my sister (Mervat) have been supportive throughout graduate school and beyond.

I extend heartfelt respect and gratitude to my lab mates: Mirzad Mohandespour, Sharhabeel

Al-Nabelsi, Hisham Almasaeid, Mohamad Saleh, Abdullah Almasoud, and Tasneem Jawadwala. Advises and comments throughout the lab discussions and during group meetings were very helpful. Also, my deepest thanks go to all of my friends in the Ames and ISU community, with special thanks to Naeem Oudat who was very supportive during all of my PhD study.

Last but not least, gratitude is also extended to the University of Jordan which financially supported my PhD study, and to Iowa State University and its staff for letting me study in such a glorious scientific edifice, and for introducing the facilities which made my study easier.

ABSTRACT

The motivation behind cognitive radio networks (CRNs) was to increase the utilization of the underutilized wireless spectrum bands. An important factor to achieving this goal is fast sensing, because if the cognitive radio (CR) node has one transceiver for sensing and transmission, then the longer the sensing time, the less the transmission time left and the lower the wireless spectrum band utilization. On the other hand, in CRNs, licensed users, also called the primary users (PUs), allow CRs to use their licensed spectrum bands provided that no harmful interference to the PUs occurs. Since there is no cooperation between the PU and the CR node, the CR node should perform periodic sensing (monitoring) to avoid interfering with the PU for more than the maximum PU's tolerable interference delay (TID). If a PU is sensed to be active, the CR node should perform out-of-band sensing (search) to find an available channel. Fast search enhances the CR node's quality of service because the CR node does not need to stop long time due to the PU appearance.

To avoid harmful interference to the PUs, monitoring and searching should be reliable enough. Higher reliability requires more accurate sensing which is achieved by using more sensing time, which leads to decreasing spectrum utilization. Therefore, there is a tradeoff between the detection speed and the reliability of sensing.

In this thesis, we study this tradeoff and propose strategies to optimize the monitoring time which is the periodic sensing time required to protect the PU from interference. Also, we optimize the search time which is the time until finding an available channel to be used by the CR nodes. In addition, we introduce a framework for cooperative in-band sensing (monitoring) that allows multiple CR nodes to share a channel, such that the channel utilization is enhanced and the sensing efficiency is increased. We propose a new definition of sensing efficiency, which

is the ratio of the size of transmitted data in one cycle to the size of the data that can be transmitted in the same cycle if there is no need for sensing.

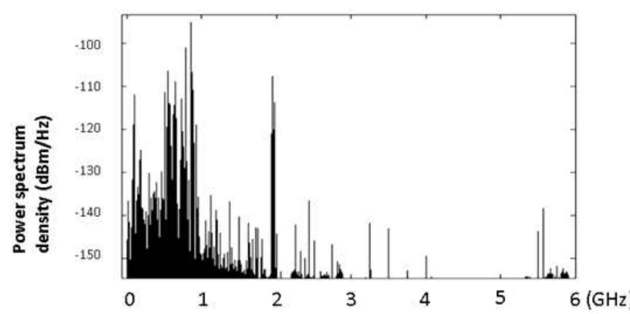
Sensing in CRNs is a key enabling functionality for the reasons mentioned above, as well as because most other functionalities in CRNs are dependent on sensing. Therefore, any function in CRN must consider sensing. Consequently, we propose a spectrum decision framework that can be used by existing routing protocols in order to enhance the throughput of a given end-to-end path, and to increase the probability of finding an end-to-end path.

In addition, we propose a cross layer routing protocol which has cooperation between the network and physical layers. Network layer finds the relay nodes jointly with the channels to be used on each hop, based on spectrum availability information which is generated by the physical layer. Both the spectrum decision framework and the cross layer routing protocol consider the monitoring time overhead of the channels, and generate recommendations to the physical layers of some CR nodes in order to sense some certain channels to enhance the quality of the selected route.

We did extensive simulation for our work: first, we show that the proposed framework of in-band sensing and channel sharing achieves better sensing efficiency than the approaches which perform periodic in-band sensing. Second, the results of monitoring time optimization and search time optimization appear fast due to the convexity of the formulations, and the time of monitoring and search is less when we relax the false alarm probability while protecting the PU. Third, the proposed spectrum decision achieves enhancement to existing routing protocols as high as 100% in some cases. Finally, the proposed routing protocol achieves better stability and throughput than existing routing protocols and increases the probability of finding a path.

CHAPTER 1 Introduction

Wireless spectrum is highly underutilized, where significant parts of it are used only for some time periods in an on/off manner and with large geographical variations. Figure 1.1 shows the measurements performed by Berkeley Wireless Research Center on spectrum band 0-6 GHz. Such low utilization in spectrum usage resulted from currently deployed static frequency allocation policy. Apparently, in order to increase wireless spectrum utilization, more flexible spectrum management techniques are required like Opportunistic Spectrum Sharing (OSS), where unlicensed users, also called secondary users (SUs), are allowed to operate in licensed frequency bands without the permission of the licensed users, also called primary users (PUs), provided that they do not introduce harmful interference to the PUs.



Freq (GHz)	0~1	1~2	2~3	3~4	4~5	5~6
Utilization(%)	54.4	35.1	7.6	0.25	0.128	4.6

Figure 1.1: Measurement of 0-6 GHz spectrum utilization at Berkeley Wireless Research Center.

The enabling technology for OSS is cognitive radio (CR) which enables the SU to sense the channels and adapt its transmission characteristics accordingly [2], using software defined radios (SDRs). In SDRs, the components that are traditionally built in hardware (e.g., modulators/demodulators, amplifiers, filters, detectors, etc.) are programmable on a personal computer or embedded computing devices. Throughout the thesis, we use SU and CR interchangeably.

There are four main functionalities in CRNs [1] as Figure 1.2 shows [1]. These are: spectrum sensing, Spectrum decision, spectrum sharing, and spectrum hand off. Spectrum sensing is required to find an available channel to use and to prevent interfering with the PU after accessing the channel. Spectrum sensing is required because there is no cooperation between the PU and the CR node.

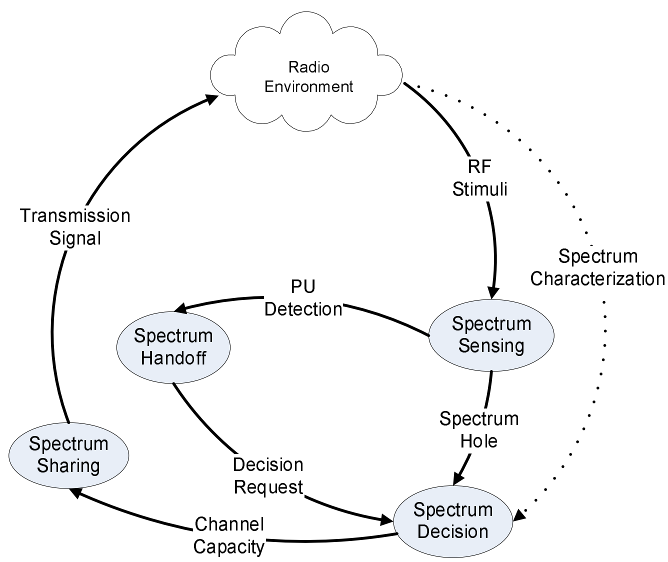


Figure 1.2: Main functions in CRNs [1]

Spectrum decision is required to decide which channel to access. It is composed of three main steps: 1) spectrum sensing, which is explained above to investigate the characteristics of the channels; 2) spectrum analysis which includes studying the characteristics of the channels, e.g., which channels maximize the throughput, minimize the delay, minimize loss rate, or achieve less access time in multi-hop routing; and 3) deciding which channel the node will use based on a specific objective and based on the application the CR node is working on. For example, the CR node may need a channel that stays available for long time regardless of the throughput, or it may need a channel that has low access time which is shared by less number of nodes, and so on.

Spectrum sharing happens when the CR node senses some channels and decides to access one of the channels. This channel may be busy by other CRs, and since the CR nodes cannot have exclusive access rights to that channel. Therefore, the CR node should share the channel with the other CRs. Sharing the channel can happen in centralized way, where there is a coordinator entity, or it can happen in distributed way which is more difficult. There is another less popular method of spectrum sharing, which is sharing the channel with the PU, when the PU does not need the channel all the time, the PU can lease the channel part of the time. Or when the PU can tolerate interference up to a specific level (interference temperature or underlay networks).

Spectrum hand-off refers to the process when the CR node needs to stop using a channel when the licensee PU becomes active. The CR node will try to find another available channel, and will inspect (spectrum sensing) multiple channels until it decides (spectrum decision) which channel to switch to. After decision, if the channel is used by some other CRs in the same area, then the CR node should share the channel (spectrum sharing) with other CR nodes. Spectrum hand-off is highly dependent on the PU behavior.

From the above exposition, it is clear that the four functionalities in CRNs are dependent on each other. However, spectrum sensing can be considered as a key enabling functionality. This is because the decisions of using and vacating channels are based on sensing results. Also, reliable sensing can help avoids interference with PUs, which is a condition to use the spectrum band of the PU.

Moreover, multi-hop routing in cognitive radio ad hoc network assumed to be cross layer approach, where the routing decision performed by the network layer, is based on channels availability found by spectrum sensing, which is done in the physical layer. Each CR node typically maintains a subset of the candidate channels, where it monitors (sense) them periodically. The routing protocol finds the path based on these subsets. However, none of the existing routing protocols, to the best of our knowledge, elaborated on how to find the subset. Indeed, the larger the subset, the better the achieved route quality, and the higher the probability to find an end-to-end path. On the other hand, the larger the subset, the more monitoring time per cycle, which implies less time for transmission and lower route throughput. Therefore, sensing time has strong effect on routing quality.

1.1 Spectrum Sensing

Spectrum sensing refers to the process of sensing a channel and deciding the state of the channel, where it can be in one of two states: 1) $\mathcal{H}0$ which means that the PU is idle, and the CR node can use the channel, and 2) $\mathcal{H}1$ means that the PU is active, and the CR node cannot use the channel.

Such decision is subject to two types of errors; false alarm and miss detection. False alarm refers to the CR detecting an active PU while the PU in inactive. Higher false alarm probability (P_f), reduces spectrum utilization by the CR. However, having less strict requirements on the false alarm has several advantages: (a) decreasing miss detection probability (P_m) due to the tradeoff between the detection probability ($P_d = 1 - P_m$) and P_f, (b) sensing potentially becomes less complex, and (c) less required sensing time for sensing a channel. Miss-detection means that the CR node detects that there is no active PU while there is an active PU. Higher miss detection probability ($P_m = 1 - P_d$), increases the interference to the PU which reduces sensing's reliability, and should be avoided.

Throughout this work, the two most popular sensing methods will be used, which are energy detection and feature detection. In energy detection, the energy in the received waveform over an observation interval (sensing time) is measured [3], and compared to a threshold value (γ). Energy detection is fast and more commonly used, but it has bad performance under low SNR. Moreover, it cannot distinguish the source of the signal whether it is from the PU or from a CR node. Therefore, in energy detection, nodes have quiet period during which, no CR is allowed to send, and sensing is performed during these quiet periods. On the other hand, feature detection identifies the existence of the PU by searching for some cyclostationary features of the PUs like modulation type and pilot signal in the received signal. Feature detection is more complex and takes more time, but does not require quiet periods and has good performance under low SNRs.

Spectrum sensing can be classified into two main types [1]: out-of-band sensing which is searching for an idle channel by sensing multiple channels sequentially until finding an available one (spectrum hole), and in-band sensing which is monitoring a channel periodically while using it, in order to prevent interference with the PU. In-band sensing requires the CR node to stop its transmission periodically to do sensing. This is because if the CR node has one transceiver, then it cannot sense and transmit simultaneously. Figure 1.3 shows the concept of a spectrum hole. During the periods marked 1, 3, and 5, the CR node stops every cycle to perform in-band sensing. Arrows 2 and 4 show switching to a different frequency band, which happens after the PU becomes busy on the current frequency band, and after the CR node performs an out-of-band sensing to find spectrum hole.

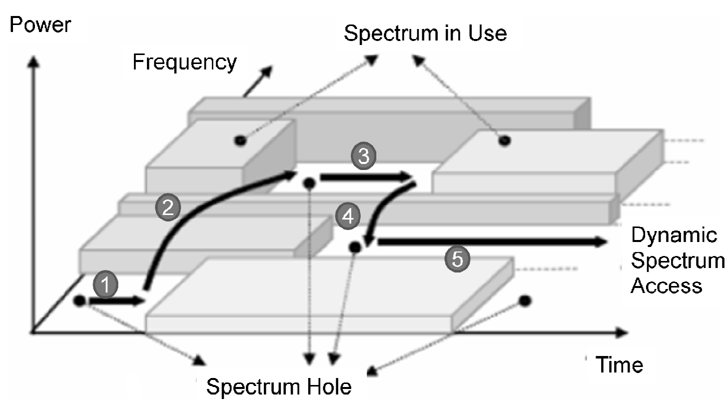


Figure 1.3: Spectrum hole concept

Usually, the PU can tolerate interference for a certain period of time which is called the tolerable interference delay (TID). When a CR node is using a channel, it should stop its transmission at least once every TID seconds and monitor PU appearance. If the PU became active, the CR node should leave the channel, perform spectrum hand-off, and search for an idle channel.

There is a need to minimize both the searching and monitoring times. This is because if the CR node has only one transceiver which is used for sensing and transmission, and if the required periodic monitoring time is ST, the room left for transmission every TID seconds is $TID - ST$. Therefore, by reducing monitoring time, room left for transmission increases, which means that the channel utilization increases. On the other hand, minimizing the search time, means when a PU becomes active, the CR node needs less time to search for an idle channel, which enhances the CR node's quality of service (QoS).

Another requirement for spectrum sensing is that it should be reliable, which implies that: 1) interference with the PU should not last for longer than TID seconds, and 2) the probability of interference with the PU should not exceed a certain probability which is defined by the PU itself.

However, having fast and reliable spectrum sensing is challenging because it involves balancing a tradeoff between the quality and the speed of sensing. Therefore, several algorithms were developed to optimize required sensing time. These algorithms can be divided into four main axes: firstly, optimizing the detection probability and false alarm probability [4]; secondly, reducing inter-sensing time in case of monitoring [5, 6]; thirdly, reducing required monitoring time [7, 8]; and finally, reducing search time [9, 10].

1.1.1 Cooperative Spectrum Monitoring

Generally, existing monitoring algorithms adopt periodic sensing, where CR nodes employ a periodic detection cycle divided into sensing and transmission times. There is a tradeoff between sensing time length and throughput: increasing sensing time will increase detection probability and reduce false alarm probability, but will also reduce transmission time. On the other hand, reducing sensing time increases transmission time, but also increases false-alarms which results in a higher number of unnecessary channel evacuations, thereby reducing the average throughput. Therefore, there is a need to select the best sensing time that protects the PU and increases transmission time.

Long monitoring time is also necessary to prevent hidden terminal problem that results from multi-path and deep shadowing. For this reason, CRs must be far more sensitive than

PUs (by 30-40 dB [11]). This requires the CR to do monitoring (or in-band sensing) for a longer time. For example, when the PU's required detection probability is 0.999, the best channel efficiency (the ratio of transmission time to cycle length) that can be achieved is only about 27% [8]. Therefore, in-band sensing time forms a non negligible overhead which reduces spectrum utilization.

One way to overcome the problem of low spectrum utilization due to long sensing time is by cooperative sensing. Cooperative sensing also prevents the hidden terminal problem, which cannot be solved by only increasing sensing time when signal to noise ratio (SNR) is below a value called "SNR wall".

Many approaches have been proposed thus far to reduce in-band sensing (monitoring) time. They still require the node to do periodic sensing like [12], [7], and [13]. The required CR sensitivity and hence required sensing time is determined by SNR, number of cooperating nodes, shadowing relationship between the cooperating nodes, and distance between cooperating nodes.

In Chapter 3, we introduce a cooperative in-band sensing framework, in which the CR nodes have two modes of operation, sensing and transmission. The nodes in transmission mode do not perform any sensing, while the nodes in sensing mode do feature-detection sensing for the channel most of the time, and they inform the nodes in transmission mode when the PU appears by flooding warning messages. Therefore, the transmission time of nodes in transmission mode is increased, and hence their throughput is enhanced. Nodes in sensing mode use feature detection sensing because the nodes in sensing mode are sensing most of the time. Therefore, they do not have a problem with slower sensing.

The advantages of this framework are summarized in two folds. First, the nodes that have data to send stop periodically to listen to warning messages generated by nodes in sensing mode. The required listening time is much shorter than the required sensing time, and it is less affected by the required sensitivity parameters. Second, our sensing framework does not require a Common Control Channel (CCC), which is used in most cooperative sensing algorithms to exchange sensing information. We prove by mathematical analysis and extensive simulations that our sensing framework enhances spectrum utilization by assigning the nodes in transmission mode higher data rate and not forcing them to stop long times to perform sensing. This enhances sensing efficiency and throughput where the nodes in transmission mode finish transmitting their data faster.

1.1.2 In-Band and Out-of-Band Sensing Optimization

Out-of-band sensing, also called searching composed of two main steps: 1) finding the sequential order of channels to be searched, and 2) Sense the channels sequentially. The sensing step can be divided into two main steps also: a) sensing the channel, and b) switching from one channel that was sensed and found to be busy, to another channel to sense it. This switching time depends on the distance between the central frequencies of the two channels and on technology factors. Therefore, the total search is dependent on: the selected sequential order of channels, the sensing time of each channel, and the switching time between channels.

The sequential order can be: a sequential order of channels, based on the decreasing probability of the channel being idle, or based on another criterion, e.g., to order channels according to increasing transmission energy. Sorting based on the probability of the channel being idle, reduces the expected number of channels to inspect until finding an available channel, which reduces the search time. However, we may select channels that are far away from each other, which increases the switching time and hence increases the search time. On the other hand, searching channels sequentially may reduce the switching time which reduces the search time. But, the CR node may need to inspect more channels to find an idle channel which increases the search time.

In Chapter 4, a heuristic solution is proposed to find a sequential order of channels to be followed during search, which reduces the search time. Also, we present two optimization formulations: one for search and the other for monitoring. The sensing in this chapter is single node sensing, where the node takes a decision by itself only. And the used underlying sensing method is energy detection because it is the fastest, and the goal is to minimize sensing time.

Our work in this chapter differs from previous work done on optimizing monitoring and search time in: 1) in monitoring, we find the sensing time jointly with the detection threshold such that the PU is protected and the sensing time is minimized. We relax false alarm probability by considering the search time as a cost for the false alarm; 2) in search, we increase the degrees of freedom where we jointly find sensing time of each channel, energy detection threshold of each channel, and the number of channels to be sensed; and 3) in both optimizations, we relax the false alarm probability. Usually false alarm probability is required to be small enough. But, we proved that in some cases sensing with higher false alarm probability requires less sensing time while achieving the required detection probability.

Another difference from existing research is the PU model. Most of the current methods use only simple partially observable Markov decision processes (POMDPs), where each radio channel is modeled with two states: Busy and Idle states [6, 9, 12, 14]. Such limited channel models do not allow the CR node to benefit from the measurements done in the last monitoring cycles. For example, in monitoring, the CR node has a memory of the last monitoring cycles and it knows that the channel was idle. But, the POMDP model is memoryless, which means that the probability that the channel is idle/busy in the current cycle is independent from the observation in the previous cycles. In Chapter 4, we model the PU idle state using multiple idle states instead of just one idle state. This allows the CR node to benefit from the previous sensing decisions done in the last monitoring cycles.

1.2 Spectrum Decision in CRNs

The use of software defined radios (SDRs) and the requirement of protecting the PU from any interference have introduced new challenges: First, the SDR allows the CR node to operate over wide spectrum bands with different characteristics. Second, if the PU can tolerate interference up to TID seconds, then the CR should sense the channel at least once every TID seconds. If sensing time takes ST ms for one channel and when the CR switches from one channel to another channel to start sensing, it takes SW ms, then a CR must spend $ST + SW$ ms for sensing each channel. This means that if the CR node has only one transceiver and is maintaining a list of K channels, then $K * (ST + SW)$ ms is wasted on sensing, which may be a significant fraction of the TID seconds cycle.

Therefore, based on the above, it is evident that choosing the best set of channels can minimize the overhead time (sensing plus switching time), which is known as spectrum decision. The CR node should know which channels it must monitor, where the set of such channels depends on the objective. For example, if the objective is optimal routing, then the spectrum decision aims to find the channels that will minimize the end-to-end delay or maximize the throughput. If the objective is to increase route stability, then the node can select the channels that are expected to be available for longer times. In Chapter 5, we will consider the spectrum decision problem where the objective is routing.

Routing in CRNs jointly selects the path and the channel to be used on each hop according to a quality objective. Quality objectives are classified into: minimizing end-to-end delay, maximize throughput, minimize interference, and increase path stability. According to [15], routing can be classified into two main types: full spectrum knowledge and local spectrum knowledge. The full spectrum knowledge assumes that there is a central entity that knows all the available channels at each CR node without sensing, thanks to spectrum availability databases [16]. Indeed, this increases the options and gives better routing results. However, as explained in [15], this is not practical.

The local spectrum knowledge approach is more practical. To the best of our knowledge, all local spectrum knowledge approaches assume that each CR maintains a set of available channels which is obtained by sensing. Then, CRs apply their routing algorithm which finds the path and the channel to be used on each hop such that their quality objective is optimized, and the channel used on each hop, should exist within the set of available channels at both nodes at the two ends of that hop.

To implement these approaches, some questions must be answered: 1) what is the optimal size of the set that should be maintained by each CR node? From the routing point of view, the bigger the set, the better the achievable routing quality objective. However, from sensing point of view, the smaller the set, the less the sensing time overhead, 2) since, as we mentioned previously, monitoring all the channels consumes considerable time which is also infeasible, then, is the routing decision that was made the best? What if there is another channel that is not used by a PU (available) and enhances the quality objective, but was not selected because the CR is not aware of its availability? and 3) if applying a specific routing algorithm was not able to find a path because, for example, on one or more hops there is no common available channel that is within the set of available channels at both nodes at the two ends of that hop: is there a possibility that there will be another channel that is available at both nodes, but the nodes are not aware of it because they did not sense it?

The objectives of the work in Chapter 5 can be summarized in: 1) introducing a new framework for spectrum decision which increases the options for a CR node by allowing it to inspect more channels, including the channels that the CR is not aware of their availability. The selection will be according to a specific criterion that takes into consideration the sensing time, the switching time, the access and channel sharing time, and the expected available channel time, 2) use this framework to enhance the performance of the existing routing algorithms, for example, by finding another channel on one hop that increases the throughput or that minimizes the end-to-end delay and 3) if applying the routing algorithm was not able to find a path from a source to a destination, we will use the framework to try to find a path because the proposed framework increases the probability of finding a path since it inspects more channels.

1.3 Routing in CRNs

While the CRN uses a wide spectrum band which spans many channels, the CR node cannot perform periodic monitoring for all the channels. The CR node needs to maintain (periodically sense) more than one channel, in order to have a backup link when the link fails due to PU appearance. In addition, one channel per node will not, most probably, provide an end-to-end path. Therefore, each CR node maintains a sub-set of the channels and each channel among this sub-set will be monitored periodically by that node. In Chapter 5, we show how to enhance the quality of a given end-to-end path, and how to increase the probability of finding a path. In Chapter 6, we will show how to design a cross layer routing protocol (CLRP).

Most of the existing routing protocols do not consider monitoring time as overhead throughout their work. Also, none of them, to the best of our knowledge, discusses specifically how the subset of channels at each CR node was selected. Reference [15] considered such routing protocols as untrue cross layer protocols, because the network layer does not tell the physical layer which channels to sense, while the physical layer only tells the network layer which channels are available.

In Chapter 6, we propose to take a broader view and consider all channels to be in the set of candidate channels to be used. We assume in CLRP that each CR node is maintaining a small set of channels which monitors them periodically. Channels that are maintained by the node are known for sure to be available. Other channels that are not maintained by the node will be considered available with certain probabilities. Therefore, we introduce a probabilistic routing approach that finds a multi-hop path between a source and a destination. The approach considers all the channels at all nodes in the network whether they are known to be available or not. If they are not known to be available, the probability of availability will be considered. We will use this approach to find end-to-end paths with enhanced throughput and stability. It will be shown by simulation that this approach achieves better throughput and longer stability than traditional approaches which only consider the channels that are known to be available at the nodes. Moreover, we will show by simulation that our approach increases the probability of finding an end-to-end path.

1.4 Thesis Contributions

In this Thesis, we introduce five contributions:

1. we propose a framework for cooperative in-band sensing which allows multiple nodes to share a channel such that the sensing efficiency is enhanced;

2. we formulate a convex non-linear formulation to optimize the required periodic sensing time (monitoring) which is required to protect the PU from interference;

3. we formulate a convex non-linear formulation which minimizes the time to search for an

available channel which is required in case of spectrum hand-off;

4. spectrum decision is proposed with the objective of enhancing the route quality of a given multi-hops path, and increasing the probability of finding a multi-hop path, while taking sensing time into consideration; and finally,

5. we propose a cross layer routing protocol, where the routing protocol selects the channels to be sensed, hence enhancing the routing quality.

1.5 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, a survey of related work is presented. Chapter 3 discusses the details of a proposed cooperative framework for in-band sensing in CRNs. Two efficient optimization formulations for spectrum searching and monitoring are introduced in Chapter 4. In Chapter 5, a spectrum decision framework that enhances routing protocols quality is proposed. After that, Chapter 6 discusses a true cross layer routing protocol in CRNs. Conclusions and some future work directions are stated in Chapter 7.

CHAPTER 2 Related Work

In this chapter, we survey the related work. First, we survey the literature on PU monitoring and in-band sensing. Then, work related to monitoring and search optimization is reviewed. Finally, the state of art work on routing and spectrum decision in CRNs is discussed.

2.1 Monitoring

References [17] and [6] focused on determining optimal transmission time, which is the cycle length minus the monitoring time. IEEE 802.22 standard [18] considers periodic in-band sensing, using both fast sensing and fine sensing. Fast Sensing typically done very fast (under 1ms). If during the fast sensing stage it is concluded that energy in the affected channel is always below the threshold, the base station may decide to cancel the next scheduled fine sensing period. Fine sensing is required based on the results of the fast sensing. During fine sensing stage, more detailed sensing is performed on the target channels.

The authors in [19, 20] showed that required monitoring time that achieves PU's required detection probability varies from node to node. Therefore, cooperative sensing was proposed in [11, 21], which achieve better detection probability with shorter times.

There is a limit on the achieved gain from increasing the number of CR nodes performing cooperative sensing [19]. Therefore, there is an optimal number of CR nodes which perform cooperative sensing such that the sensing time is minimized. We refer to this optimal number of nodes as the sensing limit. This is the least number of nodes that must be in sensing mode to counter interference constraints.

Some algorithms assume listen-before-talk strategy where the CR senses the PU channel for a certain amount of time before each packet transmission, e.g., [8], [22], and [14]. This represents a significant overhead, especially when sensing takes a long time. For instance, in IEEE 802.22, the fine sensing time is 25ms for field-sync detection [18]. This sensing time is long when compared to millisecond packet durations.

One sensing approach which assumes two modes of operation like the proposed approach exploits a separate sensor network beside the CRN. Reference [23] discussed the deployment of a wireless sensor network (WSN) to detect primary receivers, by detecting the power leaked from the local oscillator (LO) of the primary receivers. In this case, sensor nodes need to be placed in close proximity to the primary receivers. These sensors could detect the exact channel that a PU uses and transmit this information to the cognitive radios through a common control channel (CCC). The problem in this scheme is that it needs the deployment of another network, which is costly. It also depends on the weak power leaked from the LO, which requires sensor nodes to be installed close to the PU (within 1 m). Moreover, this approach does not succeed with all types of PUs, for example, mobile PU or PUs that do not leak power when they receive the signal.

The authors in [24] proposed the idea of some nodes perform sensing and feed the sensing decision to some other nodes. They introduced a performance measure called detection efficiency which is the proportion of the remaining resources that can be used for data transmission after the sensing process. This algorithm works with centralized CRN approach. Also, they need a CCC to send the cooperation results. Moreover, the CR nodes must wait for permission from the central controller to decide whether the node can transmit or not. Although our approach in Chapter 3 has some similarities, we do not require a CCC, nor we do need a central controller, and any node can switch to transmission mode depending on whether it has data to send or not, while preserving the sensing limit constraint.

Liu et al [25] discussed the ESCAPE algorithm that vacates a channel if some nodes detect a PU. They assume that not all the nodes can have the same sensing efficiency at the same time. Consequently, some nodes may detect the PU while not others. These nodes will flood their group with N predefined CDMA warning messages. They do not require a CCC, where the nodes that are transmitting must stop every cycle to listen to warning messages and to perform sensing. Still every node has to do sensing periodically. Our work introduces similar evacuation cooperation without the need for a CCC, and without forcing the transmitting nodes to do periodic sensing.

2.2 Monitoring and Search Times Optimization Related Work

Sensing time optimization usually introduces a tradeoff between protecting the PU and enhancing the performance of the CRNs. Since increasing sensing time reduces the quality of service (QoS) for the CR node, many algorithms have been developed to minimize sensing time. These algorithms can be categorized into four major classes with four objectives: Firstly, optimizing P_d and P_f such that the performance of the CR nodes are optimized [4]. Secondly, reducing inter-sensing time while monitoring [5, 6]. Thirdly, reducing the required monitoring time [6–8]. Lastly, reducing search time [9, 10, 26].

In [4], the total utility of primary and secondary systems is maximized, where the optimal threshold was found to optimize P_d and P_f. Reference [6] studied the tradeoff between sensing time and throughput. A parameter to their optimization problem was the sensing time. They find, for a given sensing time, the optimal value for the detection cycle length so that the throughput of the CR network is maximized, and the miss detection probability is not greater than a threshold. Reference [8] explored minimizing monitoring time to improve channel efficiency. None of the aforementioned algorithms and the ones that will be mentioned later considered the multi-idle states of the PU.

The work that is the closest to ours is in [7], which studied both optimizing monitoring and search times. In particular, the optimal sensing times for channel-search and channel-monitoring were obtained in a way to maximize the average throughput of the CR node while protecting the PU from harmful interference. Only one channel was considered, and it derived the optimal sensing time for a channel given the energy detection threshold. The energy detection threshold, the number of channels, and the false alarm probability were not considered in minimizing the sensing time. Also, this work does not consider channels with different characteristics.

The authors in [26] introduced the multi-band joint detection framework for wide-band spectrum sensing in a single CR. They jointly optimized a bank of multiple narrow-band detectors to improve the aggregate opportunistic throughput of a CR system while limiting the interference to the PU. They formulated the design of wide-band spectrum sensing into a class of optimization problems. They developed search time optimization problem that finds the optimal thresholds for the sub-bands in order to collectively maximize the aggregate opportunistic throughput subject to some interference constraints for each PU. In Chapter 4, we find the required sensing time for each channel and the optimal number of channels to be sensed. Also, we do sequential sensing instead of wide-band sensing which increases the granularity control.

Another trend to minimize the search time is by optimizing the order of the channels to be searched. Kim and Shin [9] introduced a sensing-sequence that sorts channels in descending order of the probability of being idle. The work in [10] finds a search sequence that helps finding spectrum opportunities with minimal delay. To achieve its goal, [10] maintains two channel lists; back-up channel list (BCL) and candidate channel list (CCL). However, they do not optimize the sensing time per channel.

Some other approaches try to minimize P_f. For example, reference [13] formulated the problem as minimizing the probability of false alarm under the constraint of probability of detection. P_f wan not relaxed and a known threshold value was assumed. Reference [27] finds the achievable minimum probability of false alarm through cooperative sensing, given a target probability of detection. In our work, we show that relaxing P_f could enhance monitoring and search time such that the PU is protected. The work in [6] studied how to select monitoring time which maximizes the achievable throughput of the CR nodes under the constraint that the PUs are sufficiently protected. They found the sensing time, and for that sensing time, they found the threshold.

In [5], a large-scale measurement-driven characterization of primary usage in cellular networks was conducted. They optimized the inter-sensing time, derived a formula for optimal inter-sensing time, and showed that large variations in inter-sensing time exist for different PU and different detection required probabilities.

2.3 Spectrum Decision and Routing in CRN Related Work

Routing decision in cognitive radio network includes deciding jointly the relay nodes and the channel to be used at each node. Reference [28] showed that separating these two steps may result in not finding a path or in degrading the performance. For this reason, most routing protocols in the literature consider joint selection of relay nodes and the channels at each hop [28–32].

Also, routing in CRN requires spectrum awareness, where the nodes should have local knowledge about the available channels at the node. Therefore, routing in CRN requires cross layer design, where route decision that is done in the network layer should be based on the channels availability collected by the physical layer through sensing. Work in [28–32] consider themselves as cross layer routing protocols.

The quality of the route depends on the set of available channels. Routing protocols in CRN can be classified into full spectrum knowledge and local spectrum knowledge [15]. In full spectrum knowledge like [33, 34], there is a central entity that has all the information about all the channels and their availability, thanks to the wireless spectrum databases [16]. These approaches, if solved optimally, should give the optimal results since they build their routing decisions based on information about all the channels without the need for sensing the channels. But according to [15], these approaches are not practical.

Local spectrum knowledge approaches assume that each node has some local knowledge about the available channels built through sensing. For example, [28,30] tried to maximize the throughput. The authors in [31, 32] tried to minimize end-to-end delay. The authors in [29] established robust paths in diverse spectrum conditions. However, all of these approaches assume that each node initially has a set of available channels which are determined by sensing. None of these approaches considered the sensing overhead. Therefore, there is a disconnect between sensing and routing. An approach that can be adopted is to use the wireless spectrum databases [16] in case it is available. But it is not always applicable for all PUs. Another method is to use a sensor network that performs sensing [35]. However, as indicated earlier this requires the deployment of a second network which is costly. Our approach is more dynamic and more practical and requires less overhead.

Reference [15] considered the above approaches as untrue cross layer protocols. A true cross layer protocol was defined as a protocol in which the information flows in both directions. However, the information in such routing protocols flows only in one direction, from the physical layer to the network layer, where the physical layer informs the network layer which channels are available. On the other hand, the network layer and the routing protocol do not instruct the physical layer which channels to sense. Our approach in Chapter 6 tries to close this gap by introducing a true cross layer routing protocol where the information flows in both directions. The physical layer tells the network layer about the initial available channels, and the network layer tells the physical layer which channels to sense and takes the sensing time into consideration.

From another perspective, routing in cognitive radio networks can be classified according to the quality objective that the routing protocol tries to optimize. Some protocols like the protocol in [29] try to maximize route stability. Others try to maximize the throughput [28,30]. Also, end-to-end delay is considered in some other protocols [31, 32]. However, none of these protocols considered monitoring time overhead. Also, none of them addressed how the initial set of available channels at each node was generated. Most of these approaches assume that this set of available channels is formed by sensing, but without considering the overhead.

2.4 Summary

In this chapter, we surveyed the work done in the literature on three main topics: first, in-band sensing. We discussed the problems of periodic in-band sensing approach. Then, we discussed the strategies proposed in literature to reduce monitoring time like cooperative sensing and using a sensor network. Second, we presented a survey of work done on minimizing searching and monitoring times. For search, work done focused on optimizing the sequential order of channels. In addition, work done on monitoring assumes two states for the PUs (idle and busy). Moreover, work that was done on searching and monitoring time optimization has less degrees of freedom and adds strict constraints on the false alarm probability, which requires longer sensing time. Finally, routing approaches in CRNs are always assumed to be cross layer approaches. But, we have presented the reasons which show that these approaches are not true cross layer.

CHAPTER 3 A Framework for Cooperative In-Band Sensing in Cognitive Radio Network

3.1 Overview

Many existing in-band sensing algorithms for CRNs adopted a periodic sensing/transmission architecture, where all the CR users do periodic sensing. Each periodic detection cycle is divided into two parts: sensing and transmission times. Sensing times are affected by many factors. In some worse scenarios (e.g., low SNRs), it may take more than half of the channel idle time for sensing. In this chapter, we propose a new cooperative in-band sensing framework to increase sensing efficiency and robustness. In our framework, each CR operates in one of the two modes: Transmission and Sensing. The CRs which have data to send switch to transmission mode, provided that there are enough CR nodes in sensing mode. Therefore, CRs in transmission mode do not have to do any sensing during transmission, which implies that they can send for longer times. CRs in sensing mode send warning messages to other nodes in case they detected the presence of PU. This cooperation (among CRs) is done on the same channel of transmission without the need for a CCC. Simulation and analytical results show that our sensing framework achieves higher sensing efficiency than traditional sensing approaches which require periodic sensing.

3.2 System Model

We assume that the network consists of both PUs and CRs, such that CRs can access the spectrum licensed to PUs if they do not interfere with them. We also assume that the CRs possess single transceiver for both signaling and data transmission. Our focus is on one primary channel monitoring. When the PU becomes active, it will occupy the channel and all CRs that use the channel must perform spectrum hand-off and vacate the channel.

CRs are assumed to be aware of the following properties about primary networks:

1. Operating frequency range: CR users are aware of the bandwidth and frequency range of the primary network.

2. Interference constraint: since CR users are visitors in the licensed bands and the PU can start anytime without informing the CR users, CRNs do not guarantee interference-free transmissions. Instead, CRNs exploit the interference constraint, which can be defined as the maximum tolerable interference delay (TID) that the PU can tolerate. In our framework, the nodes in sensing mode must inform the nodes in transmission mode about the re-appearance of the PU within the TID time.

The authors in [19] showed that under cooperative sensing, increasing the number of cooperating nodes increases detection probability up to a point, after which the detection probability will not be enhanced. We define the minimum number of nodes that are required to achieve the required detection probability as the sensing limit. To achieve this detection probability, the nodes must do sensing for a time equal $\gamma * TID$, where γ is the fraction of the detection cycle (TID) required for sensing in order to protect the PU. There are other algorithms that evaluated γ. Our framework succeeds if the sensing limit is less than the number of cooperating nodes. When the number of nodes is equal to or less than the sensing limit, the network can dynamically switch to the periodic sensing scenario.

Our problem can be defined as: given the required sensing fraction of time (γ), the sensing limit constraint, and the TID, we will develop a cooperative in-band sensing framework such that it: first, enhances sensing efficiency by increasing the transmission time and the data rate for the nodes in transmission mode; second, respects the TID by vacating the channel within TID; and third, achieves the cooperation without the need for a CCC.

3.3 A Framework for Cooperative In-Band Sensing

In our framework, the nodes work in one of two modes: transmission mode where the nodes transmit most of the time and do not perform sensing; or sensing mode where the nodes do sensing most of the time. A CR changes from the sensing mode to the transmission mode when the application layer has some data to transmit, and goes back to the sensing mode when the CR finishes data transmission as shown in Figure 3.1.

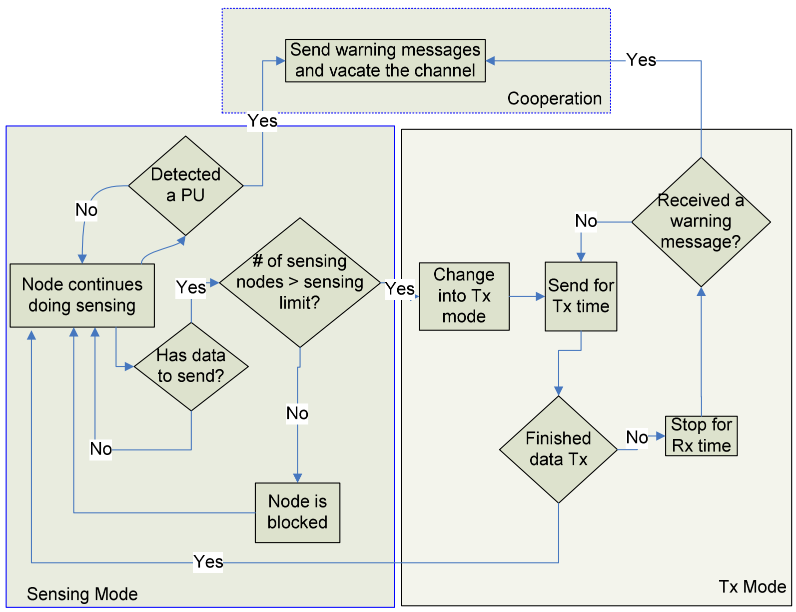


Figure 3.1: Framework for cooperative in-band sensing

3.3.1 Sensing Mode

Nodes in sensing mode conduct feature detection sensing for three reasons: first, it does not suffer from noise uncertainty that exists in energy detection. Second, it does not require quiet periods which need synchronization among CRs that use the same channel. Third, it can determine whether the signal is from a PU or from a CR. Therefore, nodes in transmission mode can transmit on the same channel that the nodes in sensing mode are sensing it. On the other hand, the long required sensing time for feature detection (85 times slower than energy detection [36]) could be overcome because the nodes in sensing mode do not transmit and can do relaxed sensing.

The main issue is to guarantee that the time for sensing, using feature detection, plus the time for sending the warning messages are less than the PU's TID, so the PU will not be interfered with for more than TID. We show later how to achieve this requirement.

In this mode, the node continues sensing the channel until a PU re-appears or until it has data to transmit. In the former case, it broadcasts a number of warning messages to the network telling them that there is a PU and this channel must be vacated. The warning messages can be sent on the same channel even though they will interfere with the PU, because the PU can tolerate interference up to TID time. In the second case, if the number of nodes in sensing mode is larger than the sensing limit, the node will switch into transmission mode. Otherwise, it will be blocked and continues in sensing mode until a node in transmission mode finishes its transmission and switches into sensing mode. Here, we do not consider the details of how to achieve the sensing limit constraint, but we suggest three approaches:

1. the first approach is that a central node coordinates the switching between the two modes for all the nodes. This node is elected dynamically according to any election algorithm [37], and can be changed with time. The coordinator node is assumed to be in sensing mode. When any sensing node wants to switch into transmission mode, it broadcasts a message to the network saying that I want to switch into transmission mode. Coordinator node checks the number of nodes in sensing mode and tells it whether it can change to transmission mode. In addition, when any node finishes transmission, it informs the coordinator node. In this case, the coordinator allows another node which was blocked due to sensing limit constraint, to switch into transmission mode.

2. a second approach could be adopted is clustering where each cluster head coordinates its own region.

3. a third approach is a random probabilistic approach where the node switches between the modes with some probability such that the sensing limit constraint is satisfied with

very high probability.

3.3.2 Transmission Mode

Nodes that have data to transmit switch into transmission mode. These nodes will not perform any sensing while transmitting where they use the sensing results of the nodes in sensing mode. Due to hardware constraints, a CR is not able to send and receive (warning messages) on the same transceiver at the same time. Therefore, it allocates part of the detection cycle (reception time) for receiving the warning messages from sensing nodes, and the remaining time for data transmission.

Reception time is much shorter than sensing time because it is less dependent on the factors which affect the required sensing time (like SNR). The other factor that affects reception time is TID, which is fixed for the same PU. For example, it is 2 seconds for the TV PU [18]. Nodes in transmission mode must stop transmission and vacate the channel within TID time.

During transmission, channel bandwidth will be shared among the transmitting nodes. For example, if the channel data rate is 6 Mbps, the cooperating nodes are 80 nodes, and the sensing limit is 20 nodes, then 60 nodes can send at the same time each with 100 kbps. This means that the nodes in transmission mode will be allocated higher data rate because the share of the nodes in sensing mode is distributed on them.

3.3.3 Cooperation Strategy

Cooperation is required for nodes in sensing mode to tell the nodes in transmission mode when a PU appears, in order to vacate the channel. We developed a method that introduces an efficient way for evacuating the used channel when some of the nodes detect the PU's appearance. It does not need any knowledge about the topology of the network because the warning messages will be flooded in the network, nor it needs any synchronization between the nodes, and can work in distributed manner.

Suppose there are N nodes in the CRN, N_t of them are in transmission mode, and N_s are in sensing mode. Suppose the largest distance between any two nodes in the network is H hops. The longest time needed for channel evacuation is when the furthest node detects the re-appearance of the PU. Assuming that a sensing node needs t_s time to detect the PU. The last node on this longest path must receive a warning message during the remaining TID-t_s time. Therefore, every CR must periodically stop transmission every t_{cycle} which is given in Equation (3.1), and listens if there is a warning message or not. It continues listening for time equals t_r seconds, therefore the node can transmit for time: $t_t = t_{cycle} - t_r$.

$$t_{cycle} = \frac{TID * (1 - \gamma)}{H} = t_t + t_r \tag{3.1}$$

For sending the warning messages, we use a method that is similar to the one used in [25]. When a sensing node detects a PU, it will start sending warning messages of length (L_w) to the group for a period equals to t_{cycle}. Between every two consecutive warning messages, there is an idle inter warning messages time of length (L_i). Before sending every warning message, it will send a prefix of length (L_p).

Every node in transmission mode must stop transmission every t_{cycle} for a reception time (t_r). During t_r, the node continues waiting for the prefix, if it received the prefix, it will wait for the warning message. In case it received the warning message, it starts sending the warning messages for t_{cycle} time, and then vacate the channel. The worst case for t_r is when the node just missed the prefix, where it must wait the next prefix. Therefore, t_r is the needed time to transmit $L_r = 2 * L_p + L_w + L_i$. Figure 3.2 shows these periods. Every node can guarantee that the neighbors received the warning messages by overhearing their forwarding of the warning messages.

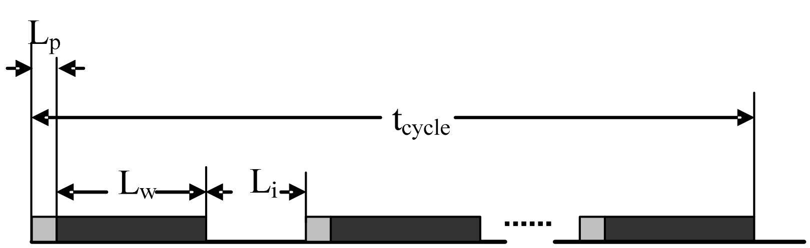


Figure 3.2: Warning messages transmission

As we can see, in our sensing framework, there is no need for a CCC since the nodes are able to communicate warning messages during the reception time (t_r), on the same channel, because during t_r, the nodes in transmission mode are listening. Also, our algorithm does not need any synchronization either between nodes in sensing mode or between nodes in transmission mode and it is distributed.

3.4 Analytical Study

In this section, we present analytical study on capacity loss, sensing efficiency, and transmission delay. We use the following notations:

- N: number of nodes in the network.
- δ: is the normalized required sensing limit.
- N_s: number of nodes in sensing mode. $N_s = \delta$ * N.
- N_t: number of nodes in transmission mode. (N_t) = (1-δ) * N.
- γ: ratio of required sensing to detection cycle.
- H: network diameter in hops
- $L_r = 2 * L_p + L_w + L_i$.
- t_r: the nodes in transmission mode must stop periodically every t_{cycle} for t_r time to receive the warning messages. t_r equals the time to send L_r bits.
- B: channel data rate.

3.4.1 Capacity Loss Analysis

First, we will show the average capacity loss due to reception/transmission cycle in our sensing framework. Second, we show the average capacity loss due to sensing/transmission cycle in traditional sensing algorithms. We normalize the capacity lost every TID seconds.

Capacity loss in our framework : for the nodes in transmission mode to receive the warning message on the same channel, it has periodic reception time which is given by the following equation:

$$t_r = \frac{L_r}{B/N_t} = \frac{L_r * N_t}{B} \tag{3.2}$$

Therefore, nodes in transmission mode must stop periodically every t_{cycle} for t_r time and can transmit for $t_t = t_{cycle} - t_r$.

Therefore, each node in transmission mode wastes time (t^O_{loss}) every TID seconds because it stops transmission in order to wait for a warning message. t^O_{loss} is given by the following equation:

$$t^O_{loss} = \frac{TID}{t_{cycle}} * t_r = \frac{H}{1-\gamma} * t_r \tag{3.3}$$

Table 3.1: Comparison between reception time and sensing time. B=6Mbps, N=100, TID=1sec, H=10hops, L_r=100bits

	traditional	Ours ($\delta = 0.1$)	Ours ($\delta = 0.5$)
$\gamma = 0.1$	100ms	16.67ms	9.3ms
$\gamma = 0.5$	500ms	30ms	16.67ms

Table 3.1 compares the reception time (in our approach) and sensing time (in traditional systems). This is the time spent by the CR node in reception/sensing per TID. It shows that the reception time is less dependent on the required detection probability and the required sensitivity, that are expressed by changing γ.

Since the number of nodes in transmission mode is N_t, and every node is assigned a data rate B/N_t, then the total capacity lost by all nodes is:

$$D^O_{loss} = t^O_{loss} * \frac{B}{N_t} * N_t = \frac{H}{1-\gamma} * L_r * N_t \tag{3.4}$$

Capacity loss in traditional sensing algorithms: every node must stop every TID seconds to do sensing for $t_s = \gamma$*TID seconds. Since all the nodes can transmit at the same time, where there is no sensing limit, the capacity lost by each node because of sensing/transmission cycles is D'^{T}_{loss}, which is given by the following equation:

$$D'^{T}_{loss} = \gamma * TID * \frac{B}{N} \tag{3.5}$$

Consequently, the total capacity lost from all the nodes (D^{T}_{loss}) in a TID period is given by the equation:

$$D^{T}_{loss} = [\gamma * TID * \frac{B}{N}] * N = \gamma * TID * B \tag{3.6}$$

Usually sensing efficiency is measured by the ratio of transmission time to the detection cycle length. Here, we develop a new sensing efficiency measure which is the ratio of transmitted data to the sum of transmitted plus lost data sizes which is given by equation (3.7).

$$\eta = \frac{D_t}{D_t + D_{loss}} \tag{3.7}$$

Table 3.2 shows a comparison between our sensing framework and the traditional sensing algorithms for different values of γ. The table shows that the sensing efficiency in our sensing framework is less affected by the required sensing time and the required sensitivity.

Table 3.2: B=6Mbps, N=100, Sensing limit=50, TID=1sec, H=10hops, L_r=100bits.

	Ours γ=0.5	traditional γ=0.5	Ours γ=0.1	traditional γ=0.1
Loss size	100Kb	3Mb	55.56Kb	600Kb
T_x Size	5.9Kb	3Mb	5.946Mb	5.4Mb
Sensing Efficiency	0.983	0.50	0.991	0.9

3.4.2 Delay Analysis

Here we will study the delay of packet transmission, supposing that initially every node has a large amount of data of size Q bits that needs to be transmitted.

Delay in our sensing framework: the nodes transmit in iterations. The process will be repeated for J ($J = \lceil (1/(1-\delta) \rceil)$ iterations. In iteration i, $N_i = min\{N_t, N-(i-1)*N_t\}$ nodes transmit their data. Therefore, the average delay for all the packets is given by the following equation:

$$D_{avg}^{\prime O} = \frac{\sum_{i=1}^{J}[((i-1)*\frac{Q*N_t}{B}+\frac{Q*N_i}{B})*N_i]}{N} \tag{3.8}$$

This is in case the nodes transmit continuously without stopping for reception time. But, the nodes in transmission mode have periodic reception time (t_r). Therefore, the actual average delay is given in the equation:

$$D_{avg}^{O} = D_{avg}^{\prime O} + \frac{D_{avg}^{\prime O}}{(TID*(1-\gamma))/H - t_r} * t_r \tag{3.9}$$

Delay in traditional sensing algorithms: in case the nodes transmit continuously without stopping for sensing, the average delay for all the packets in traditional sensing algorithms is given in the equation:

$$D_{avg}^{\prime T} = \frac{Q}{B/N} \tag{3.10}$$

But in traditional sensing case, the nodes do sensing periodically. Therefore, the actual average delay is given in equation (3.11).

$$\begin{aligned} D_{avg}^{T} &= D_{avg}^{\prime T} + \frac{D_{avg}^{\prime T}}{(1-\gamma)*TID}*(\gamma*TID) \\ &= \frac{D_{avg}^{\prime T}}{(1-\gamma)} \end{aligned} \tag{3.11}$$

Table 3.3 compares the average response time required to send 10 Mb data. The results show that our sensing framework outperforms traditional sensing algorithms.

3.5 Simulation Results

Our simulation was done on a channel with data rate 6 Mbps. We assumed that the PU has TID = 1 second. We used warning message length (L_w) = 60 bits, idle enter warning message

Table 3.3: Average response time for every node to send 10Mb using our sensing framework and traditional sensing algorithms. TID=1 second, H=10 hops, L_r=100 bits, B=6Mbps, Q=10Mb, N=100 nodes

γ	Ours δ=0.5 (sec)	Ours δ=0.1 (sec)	traditional (sec)
γ=0.5	154.24	127.11	333.33
γ=0.1	157.36	126.17	185.18

time (L_i)=10 bits, and prefix time (L_p) = 6 bits. These values are similar to the values used in [25]. We will show the effect of changing warning message size on our work.

We compared our sensing framework with the traditional sensing algorithms that do periodic sensing. Here, γ represents the ratio of required sensing time to the detection cycle length. Our sensing framework is less affected by γ. Therefore, during all the following results, we used γ=0.5 for the results associated with our sensing framework, while it was changed for traditional systems.

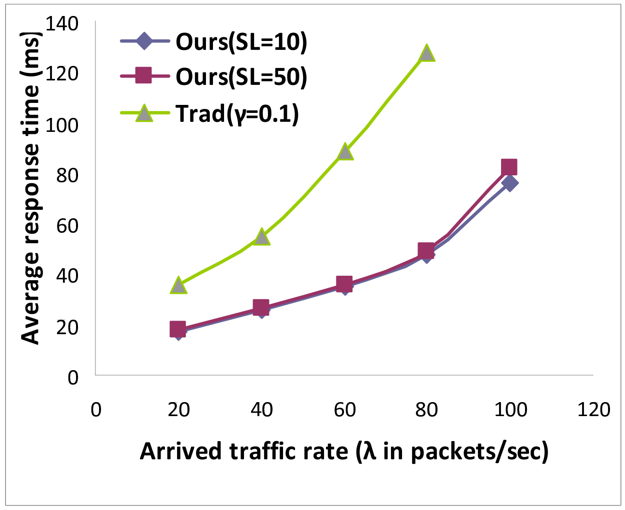
All of the simulations were done on 100 nodes. Simulation time = 500 seconds. Packets arrive to the network with Poisson distribution with rate λ and assigned to a node randomly. All the results are the average of 10-20 trials. We assumed multi-hop communications with largest number of hops is 10 hops.

Figures 3.3.a and 3.3.b compare between our sensing framework and traditional sensing algorithms. They compare the average response times versus different values of sensing limits, traffic rates (λ), and average transmitted packet sizes. We assumed a large number of hops in our framework (10 hops) and low sensing time for traditional sensing algorithms ($\gamma = 0.1$) while $\gamma = 0.5$ for our sensing framework. We assumed that the traditional algorithms will not be affected by the number of hops. These figures show that our sensing framework outperforms traditional sensing algorithms.

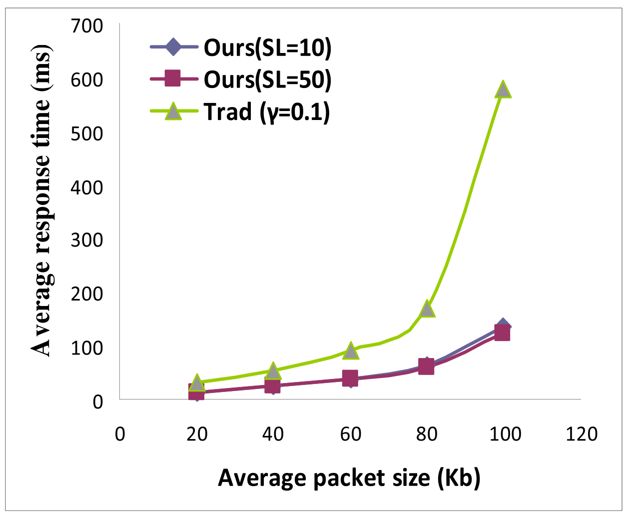
Figure. 3.3.c shows the effect of changing warning message length on the performance of our sensing framework. During the previous experiments, the reception time was the time to send 82 bits. The Figure shows that the performance of our sensing framework is slightly affected by changing the reception time, and it is still better than traditional sensing algorithms, even with low sensing ratio ($\gamma = 0.1$).

3.6 Summary

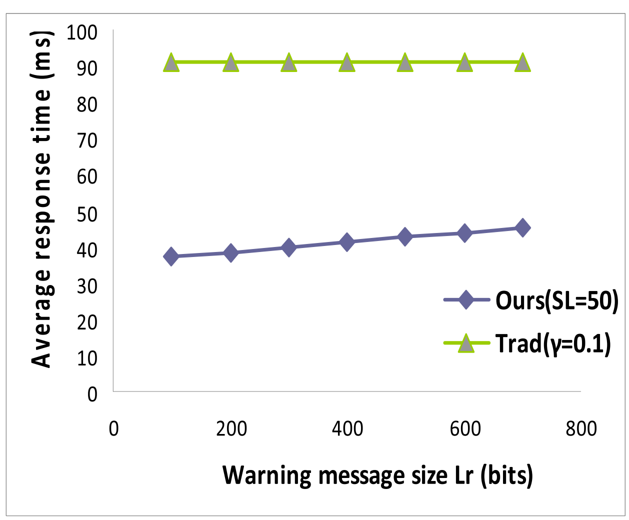
In this Chapter, we developed a new sensing framework for channel sharing. It enhances sensing efficiency for CR nodes, by allocating more time to the CRs that have data to send. In the proposed framework, CRs have two modes of operation: sensing and transmission. Nodes in sensing mode, do sensing most of the time and they inform other nodes in case they detected PU appearance. Nodes in transmission mode are transmitting most of the time, and have a periodic transmission/reception cycle, with the reception time much shorter than the sensing time. Moreover, the cooperation between nodes in sensing mode and nodes in transmission mode does not require a CCC. Simulation results show that our sensing framework outperforms traditional sensing algorithms according to the achieved sensing efficiency.



(a) Packet sizes=50Kb and hops count=10hops



(b) λ=50 and hops count=10hops



(c) Packet size=60Kb and λ=50

Figure 3.3: comparisons between our sensing framework and traditional sensing algorithms

CHAPTER 4 Efficient Spectrum Searching and Monitoring in Cognitive Radio Network

4.1 Overview

Two objectives of sensing in cognitive radio (CR) are to detect the primary user (PU) accurately and quickly, which are contradicting objectives. Therefore, many papers try to optimize this tradeoff and find the minimum sensing time which protects the PU. The trends are classified in enhancing false alarm probability (P_f) and detection probability (P_d), optimizing inter-sensing time, in-band sensing (monitoring) time optimization, and out-of-band sensing (search) time optimization.

The contribution introduced in this chapter can be summarized in four folds: 1) We introduce a new PU model which models the PU idle time into multiple idle states with certain lengths and certain probabilities. 2) We use this model to formulate a convex non-linear optimization, which finds the best sensing time, energy detection threshold, and false alarm probability of the channel being monitored. 3) We introduce a heuristic solution that tries to find the best sequential order of channels to be followed during the search such that the search time is minimized. 4) We formulate a convex non-linear search time optimization formulation. The formulation finds the best number of channels to sense, the threshold of each channel, the sensing time of each channel, and P_f of each channel such that the PU is protected, the sensing time is minimized, and the CR will find an available channel with very high probability.

Our work differs from previous work in: 1) In monitoring, we find the sensing time jointly with the detection threshold such that the PU is protected and the sensing time is minimized. We relax false alarm probability by considering the search time as a cost for the false alarm; 2)

In search, we increase the degrees of freedom where we jointly find sensing time of each channel, energy detection threshold of each channel, and the number of channels to be sensed; 3) In both optimizations, we relaxed the false alarm probability. Usually false alarm probability is required to be small enough. But, we proved that in some cases sensing with higher false alarm probability requires less sensing time while achieving the required detection probability.

Another difference from existing research is the PU model. Most of the current methods use only simple partially observable Markov decision processes (POMDPs), where each radio channel is modeled with two states: Busy and Idle states [6, 9, 12, 14]. Such limited channel models do not allow the CR node to benefit from the measurements done in the last monitoring cycles. For example, in monitoring, the CR node has a memory of the last monitoring cycles and it knows that the channel was idle. But, the POMDP model is memoryless, which means the probability that the channel is idle/busy in the current cycle is independent from the decisions in the last cycles. In this chapter, we model the PU idle state into multiple idle states instead of just one idle state. This allows the CR node to benefit from the previous sensing decisions done in the last monitoring cycles.

4.2 System Model

The CR node, while transmitting on a channel, should stop transmission periodically to sense (monitor) the channel. If during monitoring, the channel found to be busy, the CR node should search the remaining channels until it finds an idle channel. The search part usually composed of two main steps: first, sorting the channels in a way to minimize the search time; second, sensing the channels sequentially following the sequential order in the first step. The sensing time of each channel can be optimized separately such that the PU is protected, or the sensing time of a set of channels can be optimized jointly. We use the joint optimization option since it increases the degrees of freedom and gives less search time. Also, we propose a heuristic approach for sorting the channels. The sorting is iterative based on minimizing the sensing plus switching time given the current channel.

Typically, the PU is modeled as a renewal process with two states (idle and busy). This model is memoryless, which means that if the CR node knows that in the previous monitoring cycles, the PU was idle or busy, this does not add any information to the current monitoring cycle. But, if we can model the PU idle/busy states into multiple idle/busy states, each with specific length and with specific probability which can be found through long term observation, this will be useful during monitoring and search. The CR node performs monitoring every detection cycle, which means that it has a memory that the PU was idle during the last detection cycles. Therefore, a PU model with multi-idle states can be beneficial during monitoring. The multi-busy states can be beneficial during the search because it provides the CR node with the probability of the channel being busy or idle. But, since the CR does not have memory about the last detection cycles of the channel, we will not model the busy state into multi-busy states, i.e., it will be modeled as one busy state.

We use energy detection as the basic detection method. In this method, the energy in the received waveform over an observation interval (sensing time) is measured [3], and compared to a threshold value (γ). To detect a weak primary signal on specific channel, one could pose a binary hypothesis testing as follows:

$$y_j \sim \begin{cases} v(j) & \text{Under } \mathcal{H}0 \\ s(j) + v(j) & \text{Under } \mathcal{H}1 \end{cases} \quad \forall j \in [1 - N] \tag{4.1}$$

where $\mathcal{H}0$ represents the absence of the primary signal, i.e., the received baseband complex signal y_j contains only additive white Gaussian noise (AWGN), i.e., $v(j) \sim \mathcal{N}(0, \sigma_v^2)$, and $\mathcal{H}1$ represents the presence of the primary signal, i.e., y_j consists of a primary signal $s(j)$ corrupted by $v(j)$. N corresponds to the number of measured samples. Energy detection is a threshold-based hypothesis test. It means that the energy on a specific channel is measured and compared to a threshold value similar to the following hypothesis test:

$$V(\mathbf{y}) = \sum_{\mathbf{j=1}}^{\mathbf{N}} \mathbf{y_j^2} \underset{\mathcal{H}\mathbf{0}}{\overset{\mathcal{H}\mathbf{1}}{\gtrless}} \gamma \tag{4.2}$$

where $V(\mathbf{y})$ is the test statistics, N is the number of measured samples which represents the sensing time, y_j^2 is the energy measured on sample j, and γ is the energy detection threshold.

$V(\mathbf{y})$ is a random variable whose probability density function is $t_0(x)$ under $\mathcal{H}0$ and $t_1(x)$ under $\mathcal{H}1$. According to the central limit theorem, $V(\mathbf{y})$ is asymptotically normally distributed if N is large enough ($N \geq 20$ is practically sufficient). When a CR performs energy detection of the channel i, for large N_i, and when the signal to noise ratio to the PU on channel i is SNR_i, the false alarm probability, P_f^i, and the detection probability, P_d^i, can be approximated by the following two equations:

$$P_f^i(\gamma_i, N_i) = Q((\frac{\gamma_i}{\sigma_v^2} - 1)\sqrt{N_i}) \tag{4.3}$$

$$P_d^i(\gamma_i, N_i, SNR_i) = Q\left[(\frac{\gamma_i}{\sigma_v^2} - SNR_i - 1)\sqrt{\frac{N_i}{2 * SNR_i + 1}}\right] \tag{4.4}$$

where:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\tau^2/2} d\tau \tag{4.5}$$

is the tail probability of a zero-mean unit-variance Gaussian random variable.

4.3 PU Model

In this chapter, we assume that we have statistical information about the PU through long term observation. This information could be represented by radio environment maps (REM) [38–40] through accurate cooperative sensing [41]. Supposing that we are doing monitoring every $\tau seconds$, where τ is fixed and it is PU-dependent (i.e., 2 seconds for TV PU [42]), then we derive the probability that the PU will stay idle for $p * \tau$ seconds (PI_p), $\forall p \in [1, I]$.

Since in monitoring, the CR node perform monitoring every τ, then it can maintain a memory of the last monitoring decisions. This memory as we will see later affects the required monitoring time. We assume that instead of only one idle state of the PU, there are multiple idle states (I), where the PU behavior model could be represented by Figure 4.1 for the duration of idle periods. In Figure 4.1, each circle represents how long the PU is going to stay idle in terms of sensing periods (τ). For example, the circle with caption $2 * \tau$ means that the PU is going to stay idle for 2 sensing periods with probability PI_2, and then becomes busy with probability 1.

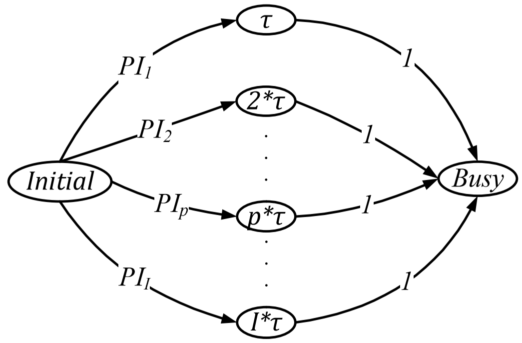


Figure 4.1: PU Model

Assuming that all probabilities in Figure 4.1 are known, then the expected length of the idle time of the PU ($\bar{TI}$) can be found by the following equation [43]:

$$\bar{TI} = \sum_{p=1}^{I} p * \tau * PI_p \tag{4.6}$$

We are assuming that PI_p is variable for different p, which means the PU stays idle for different lengths with different probabilities. The probability that the CR selects an idle period of length k monitoring cycles is:

$$\hat{P_k} = \frac{k * \tau * PI_k}{\bar{TI}} \tag{4.7}$$

We assume that the PU has I idle states, and the CR starts using the channel at arbitrary point in time. Therefore, the CR node does not know for sure in which state it is. The probability that the CR started monitoring the PU in the j^{th} monitoring cycle given that the idle time length is k monitoring cycles is $\frac{1}{k}$, where $j \leq k$. Hence, the probability of being in state p ($R(p)$) can be given in the equation:

$$\begin{aligned} R(p) &= \sum_{k=p}^{I} \frac{1}{k} * \frac{k * \tau * PI_k}{\bar{T}I} \\ &= \sum_{k=p}^{I} \frac{1}{k} * \frac{k * \tau * PI_k}{\bar{T}I} \\ &= \frac{\sum_{k=p}^{I} PI_k}{\sum_{k=1}^{I} i * PI_k} \\ &= \frac{Pr(\text{PU idle period} \geq p \text{ cycles})}{\sum_{k=1}^{I} k * PI_i} \end{aligned} \tag{4.8}$$

Figure 4.2 is more general than traditional PU model with two states, this is because it has limited number of idle states, and each of these states can be with certain probability. If all the idle times are with the same probability, and the number of states are unlimited, then the probabilities in Figure 4.2 can be derived from the two states model with exponential time. The multi-idle states are useful in monitoring, where a CR node performs monitoring every sensing period (τ). Suppose that the CR node detected that the PU was idle in the last p sensing periods. Let q_p be the probability that the PU becomes busy after p sensing periods given it was idle in the previous p sensing periods, and given that the CR started monitoring from the first idle period. Using the simplified model in Figure 4.1, q_p can be calculated using the following equation:

$$PI_p = \prod_{j=1}^{p-1} (1 - q_j) * q_p \tag{4.9}$$

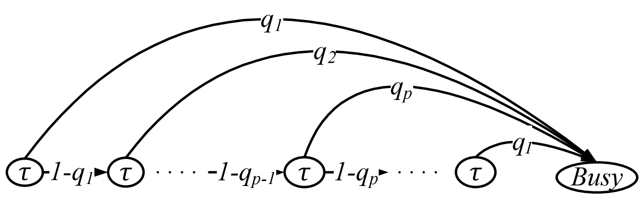


Figure 4.2: Simplified PU model

where $q_1 = PI_1$. Therefore, the probability that the PU being idle is dependent on the time since the CR has started monitoring. For example, in the first monitoring cycle, the probability to be in state p ($PS_1(p)$) can be given in the equation:

$$PS_1(p) = R(p-1) * (1 - q_{p-1}) \tag{4.10}$$

And accordingly the probability of being idle in the first monitoring cycle ($Pr^{(1)}(\mathcal{H}0)$) is:

$$Pr^{(1)}(\mathcal{H}0) = \sum_{p=2}^{I} PS_1(p)) \tag{4.11}$$

For the c^{th} monitoring cycle, the probability of being in state p ($PS_c(p)$):

$$PS_c(p) = R(p-c) * \prod_{j=p-c}^{p-1} (1 - q_j) \tag{4.12}$$

And accordingly, the probability of being idle in the c^{th} monitoring cycle ($Pr^{(c)}(\mathcal{H}0)$) is:

$$Pr^{(c)}(\mathcal{H}0) = \sum_{p=c+1}^{I} PS_c(p) \tag{4.13}$$

The problems that we address below are as follows: 1) first, given the primary SNR, PI_p $\forall\ p \in [1, I]$, the monitoring cycle that the CR is in, the required detection probability of the PU ($\bar{P}_d$), and the average search time (T_{search}), we formulate a monitoring time convex optimization formulation that finds jointly the optimal monitoring time, detection threshold, and false alarm probability; 2) Second, given the primary SNR_u, $Pr_u(\mathcal{H}0)$, $\bar{P}_d(u)$ $\forall\ u \in [1, M]$, and which channel the node was using, we heuristically find a sequential order of the M channels to be followed during the search such that the search time until an available channel is found is minimized; 3) Third, given the primary SNR_i, $Pr_i(\mathcal{H}0)$, $\bar{P}_d(i)$ $\forall\ i \in [1, M]$, and the best order of the M channels to be followed when doing search, we develop a convex non-linear search time optimization formulation that finds the sensing time of each channel, γ, and the best number of channels to search such that the total search time is minimized, PUs are protected, and the CR node will find an idle channel with high probability.

4.4 Monitoring Optimization

The goal of in-band sensing (monitoring) is to prevent interfering with the PU. Monitoring should satisfy two requirements: 1) Detection time should be less than PU tolerable interference delay (TID). Assuming the CR node is going to sense the channel every τ seconds for time t_m, then transmits for $\tau - t_m$ if the channel is idle. However, if the channel is found to be busy, the CR node will search for an empty channel for an average search time T_{search} seconds. 2) The detection probability from doing sensing (P_d or $Pr(\mathcal{H}1|\mathcal{H}1)$) given by Equation (4.4) should be greater than the given detection probability of the PU ($\bar{P}_d$).

Monitoring a channel has four candidate results:

1. $\mathcal{H}0|\mathcal{H}0$ (true positive): detects it idle while it is idle. In this case, the CR node starts sending on the channel.

2. $\mathcal{H}0|\mathcal{H}1$ (false positive or miss detection): detects it idle while it is indeed busy. In this case, the CR node starts sending, but interfering with the PU.

3. $\mathcal{H}1|\mathcal{H}1$ (true negative or detection): detects it busy and it is indeed busy. In this case, the CR node should vacate the channel, and search for another available channel.

4. $\mathcal{H}1|\mathcal{H}0$ (false negative or false alarm): detects it busy while it is indeed idle. In this case, the CR node vacates the channel, and searches for another available channel.

Note that false positives should be avoided since they result in collisions with PU transmission, and false negatives should also be avoided since they waste available transmission opportunities by CRs. Algorithm 1 describes a non-linear optimization algorithm that minimizes the monitoring time.

Algorithm 1 : Monitoring in the c^{th} monitoring cycle

1: Min $N + T_{search} * (Pr^{(c)}(\mathcal{H}0) * Pr(\mathcal{H}1|\mathcal{H}0) + Pr^{(c)}(\mathcal{H}1) * Pr(\mathcal{H}1|\mathcal{H}1))$
2: s.t. $Pr(\mathcal{H}1|\mathcal{H}1) \geq \bar{P}_d$
3: $t_m \leq \tau$

In Algorithm 1, the first line is the objective function that we are trying to minimize. It consists of three parts: 1) monitoring time (N) which will be executed in the current monitoring cycle; 2) search time (T_{search}) which will be executed in case the channel is idle, but detects it as busy (false alarm). This will happen with probability $Pr^{(c)}(\mathcal{H}0) * Pr(\mathcal{H}1|\mathcal{H}0)$, and 3) search time ($T_{search}$) which will be executed in case the channel is busy and detect it as busy. This will happen with probability $Pr^{(c)}(\mathcal{H}1) * Pr(\mathcal{H}1|\mathcal{H}1)$. In fact, Algorithm 1 tries to minimize the monitoring time which is the first part (N), but the other two parts are included in the objective function because they are the cost of doing less monitoring time which increases the false alarm probability. Without including these two parts in the objective, the monitoring time can be minimized arbitrarily because there is no constraint on false alarm probability in Algorithm 1.

The second line in algorithm 1 is a constraint to satisfy the second requirement of monitoring's requirements previously mentioned. In this constraint, PU detection probability condition is guaranteed, i.e., $Pr(\mathcal{H}1|\mathcal{H}1)$ from doing sensing must be greater than or equal to the given detection probability ($\bar{P}_d$) of the PU. $Pr(\mathcal{H}1|\mathcal{H}1)$ is given in Equation (4.4). The third line is that the monitoring time should be less than the detection period. In case when the SNR is below the SNR wall [44], the monitoring algorithm will not be able to detect the primary signal.

The decision variables are the monitoring time (N) and the energy detection threshold (γ) of the channel being monitored. The enhancement in sensing time comes from the model described in Figures. 4.1 and 4.2, and from relaxing the false alarm probability. The longer the CR is using the channel, the larger the probability the channel will be busy. Also, the larger the average search time (T_{search}), the better to do longer monitoring time to achieve less false alarm probability.

The parameters to this algorithm are: $Pr^{(c)}(\mathcal{H}0)$, primary SNR, and average search time (T_{search}) that will be needed in case the PU found to be active whether it is correct or not. SNR is the signal-to-noise ratio between the CR node and the monitored PU. $Pr^{(c)}(\mathcal{H}0)$ is calculated using the PU model through Equation (4.13), i.e., if the CR node was using the channel in the last $c-1$ periods, where it sensed the channel in the last $c-1$ periods and found the channel to be idle, then the CR node finds the probability to be idle, $Pr^{(c)}(\mathcal{H}0)$ and to be busy, $Pr^{(c)}(\mathcal{H}1) = 1 - Pr^{(c)}(\mathcal{H}0)$ in the c^{th} monitoring period.

There is a hidden convexity in this optimization under certain conditions. First, this is a minimization problem, which means that the constraints and the objective function should be convex for the optimization to be convex. The objective function is a summation of three things:

1. The monitoring time (N): it is linear which means it is convex.

2. The parameter (T_{search}) multiplied by the false alarm probability ($Pr(\mathcal{H}1|\mathcal{H}0)$): the false alarm probability is given in Equation (4.3), and it is a Q function which is convex for values ≤ 0.5. Therefore, This term is convex for false alarm probability less than 0.5, which is typically required to be less than that. Therefore, by adding a constraint ($Pr(\mathcal{H}1|\mathcal{H}0) \leq 0.5$), this term will be convex.

3. The parameter (T_{search}) multiplied by the detection probability ($Pr(\mathcal{H}1|\mathcal{H}1)$). According to Equation (4.4), and since Algorithm 1 is a minimization algorithm, the objective function will be minimum when the constraint in line 2 is equal. Therefore, the value of ($Pr(\mathcal{H}1|\mathcal{H}1)$) will be equal to $\bar{P}_d$, which means that this term can be replaced by the constant $T_{search} * \bar{P}_d$ which is convex.

The first constraint is a greater than or equal inequality. For a non-linear constraint to be convex, it should look like: *"convex non-linear terms $\leq$ constant"*. Also, since the negative of a concave non-linear term is indeed convex, then the non-linear term should be concave for the constraint to be convex. The detection probability ($Pr(\mathcal{H}1|\mathcal{H}1)$) is given in Equation (4.4). Again, it is a Q function that is concave for values ≥ 0.5 which is a desired range for detection probability, where it is typically required to be greater than 0.9. Therefore, the optimization problem is a convex optimization for $\bar{P}_d \geq 0.5$ and $P_f \leq 0.5$. This convex optimization can be solved using convex optimization algorithms that have quadratic convergence, such as Newton's algorithm and Sequential Quadratic Programming (SQP).

4.5 Finding the Sequential Order of the Searched Channels

We mean by search here the process of finding an available channel. In search, the CR sequentially sense multiple channels until finding an available channel. Minimizing the search time enhances the QoS of the CRs. This is because the CR does not need to stop transmission for a long time to find an available channel when a PU appears. Search time minimization can be done in two ways: Firstly, by finding the optimal sensing time of each channel to be searched. Secondly, minimize search time by optimizing the search sequential order of channels.

The search time is composed of two main parts: 1) sensing time of each channel to be searched, and 2) the switching time that is required in case a channel is sensed and found to be busy. In this section, we develop a heuristic strategy that finds the sequential order of the channels to be searched such that the total search time until finding an available channel is minimized. Initially, the CR is using one of the channels (f_0). This channel became busy due to PU appearance. In the next section, we find the sensing time of each of the channels.

Finding the optimal sequential order requires: 1) taking all the permutations of the channels, 2) find the search time for each permutation of the channels, and 3) select the permutation that minimizes the search time. Since there are exponential number of permutations, this solution is impractical. Therefore, we introduce a heuristic solution that finds the sequential order iteratively, one channel per iteration.

Suppose that we have M channels, the node is initially on channel f_0, and we want to find the next channels to sense, $f_s \; \forall \; s \in [1, M]$, such that $f_s \neq f_0$. Factors that affect the sensing plus switching time (search time) are: SNR_s, $Pr_s(\mathcal{H}0)$, and the required detection probability, $\bar{P}_d$(s) of channel s. During channels sorting and search optimization, the CR node does not have memory about the searched channels. Therefore, the multi busy/idle states model is not suitable. Instead, a two states renewal process model that composed of one busy and one idle state with exponential lengths will be used. $Pr_s(\mathcal{H}0)$ and $Pr_s(\mathcal{H}1)$ for the channel s can be calculated using that model. Then, to find the next channel that minimizes the sensing plus switching time, we propose the following optimization formulation:

Algorithm 2 : Finding the best first channel to sense

1: For s=1 up to M
2: Min $t(s) = [ts(s) + t_{sw}(f_0, f_s)] * Pr_s(\mathcal{H}1)$
3: s.t. $Pr_s(\mathcal{H}1|\mathcal{H}1) \geq \bar{P}_d(s)$
4: $Pr_s(\mathcal{H}1|\mathcal{H}0) \leq \delta$
5: End For
6: f(1) = f(z) such that t(z) is minimum.

in Algorithm 2, $ts(s)$ is the sensing time of channel s which is the number of samples

(N_s) divided by the sampling frequency. $t_{sw}(f_0, f_s)$ is the switching time from f_0 to f_s which depends on the two frequencies. We assume in this work that the switching time satisfies the triangularity and linearity where $t_{sw}(f_0, f_s) = \alpha * |f_s - f_0|$, where α is a technology factor.

The intuitive meaning of this optimization is: find the channel to be sensed such that the sensing plus switching time is minimized, the PU is protected, and the false alarm probability is less than small value (δ). We are including the false alarm probability due to the trade-off between the false alarm and the detection probability. i.e., not considering the P_f, we can reduce sensing time arbitrarily by manipulating the energy detection threshold while protecting the PU. Therefore, the small false alarm probability (δ) in fact will not allow the sensing time to be arbitrarily small. Including $Pr_i(\mathcal{H}1)$ in the objective makes the optimization favors channels with lower probability of being busy.

This is a non-linear optimization. But, similar to Algorithm 1, this is a convex optimization for the same reasons. Algorithm 2 finds only the next channel. In the same manner, we can find the best sequence of channels iteratively. In each iteration, we find the channel that minimizes the sensing plus switching time among the remaining channels, e.g., in iteration i, we find among the remaining $[f(i) - f(M)]$ channels, the channel which minimizes the sensing plus switching time (switching from $f(i-1)$). And in the next iteration, the channel $f(i+1)$ that minimizes the sensing plus switching time from $f(i)$ will be found. Algorithm 3 shows our approach of finding the best sequential order of channels.

Algorithm 3 : Finding the best sequence of channels

For $i = 1$ up to M
  Min=∞ , MinIndex=-1
  For $s = i$ up to M
    Min $t(s) = [ts(s) + t_{sw}(f_0, f_s)] * Pr_s(\mathcal{H}1)$
    s.t. $Pr(\mathcal{H}1|\mathcal{H}1) \geq \bar{P}_d(s)$
       $Pr(\mathcal{H}1|\mathcal{H}0) \leq 0.1$
    if ($t(s) \leq$ Min)
      Min=$t(s)$
      MinIndex=s
    End if
  End For
  Temp=$f(i)$
  $f(i)$=f(MinIndex)
  f(MinIndex)=Temp
  $f_0 = f$(MinIndex)
End For

In iteration i of the outer for loop, a channel that minimizes the sensing + switching time will be found. The inner for loop searches the $M - i$ channels to find the channel which minimizes the sensing plus switching time and makes it the i^{th} channel to be sensed. Lines 4-6 finds the minimum sensing + switching time for each channel given the current channel. Lines 7-10 keep track of the channel that minimizes the sensing + switching time. Lines 12-15 swap the next channel with the channel that minimizes sensing + switching time.

4.6 Search Optimization

In this section, we consider out-of-band sensing (search) optimization.

4.6.1 Search Definition

Out-of-band sensing (search) target is to find an available channel to use. Search is also required in case of spectrum hand-off (when the PU re-appears). Therefore, search process needs to be done very fast in order to enhance the quality of service (QoS) of the CR nodes, and for their transmissions not to be interrupted for a long time.

During the search, the PU detection probability (P_d) requirement should be satisfied. This condition is a little bit different from that in monitoring. In monitoring, the CR node monitors one channel. In search, the CR node looks for an available channel. Therefore, there are multiple channels with multiple PUs and different detection probabilities to be satisfied.

Similar to the sorting phase, the CR node does not have memory about the last detection cycles of the channels to be searched. Therefore, the PU model with multi idle/busy states cannot be used. Instead, the two busy/idle states model with exponential times is used. From this model, the probabilities of channel i being idle ($Pr_i(\mathcal{H}0)$) and being busy ($Pr_i(\mathcal{H}1)$) can be found.

In the monitoring optimization section, we optimized the monitoring time for one channel. In the previous section, we showed a heuristic method to find the optimal sequence of channels which minimizes the search time. Using the sequential order that is generated in the previous section with the way of optimizing the channel sensing time for each channel separately will reduce the search time. However, when we consider optimizing multiple channels jointly in one optimization formulation, the total sensing time of multiple channels will be less than the total sensing time when the sensing time of each channel optimized separately. Therefore, in this section, we derive a way to jointly find: 1) the number of channels to sense, 2) the sensing time of each channel, 3) the energy detection threshold of each channel, and 4) the false alarm probability of each channel such that the PUs are protected, the total search time is minimized, and the CR finds an available channel with high probability.

4.6.2 Optimization

Usually, sensing is done such that the false alarm probability (P_f) is reduced [13, 27]. However, sometimes sensing more channels with higher P_f will be better than sensing fewer channels with lower P_f. Algorithm 4 shows channels search-time optimization.

Algorithm 4 : Search Optimization

1: Min $\sum_{i=1}^{K}(ts(i) + F(\alpha, f_i, f_{i-1})) * Pr(sw)$

2: s.t. $Pr_i(\mathcal{H}1|\mathcal{H}1) \geq \bar{P}_d(i)$ for $i \in [1, K]$

3: $1 - \prod_{i=1}^{K}(Pr_i(\mathcal{H}1) + Pr_i(\mathcal{H}0) * Pr_i(\mathcal{H}1|\mathcal{H}0)) \geq \zeta$

Algorithm 4 is a non-linear programming optimization formulation. The intuition is: if the CR wants to search only K out of the M channels following the order given in the previous section, the optimization finds the sensing time of each channel, the energy detection threshold of each channel, and the false alarm probability of each channel, which minimize the total search time such that the PUs are protected against interference, and the CR node finds an available channel with high probability. The first line is the objective function which is the expected search time, and to be minimized. Line 2 means that the detection probability requirement of each PU must be satisfied. Line 3, means that the CR node will find an idle channel with a probability that is at least equals to ζ.

The objective function, which is the expected search time, is composed of the sum of the sensing time of each channel ($ts(i)$) which is the number of samples divided by the sampling frequency, and the switching delay between the channels ($F(\alpha, f_i, f_{i-1})$) multiplied by the probability of switching. The switching delay could be zero if multiple narrow-band detectors are used [26]. Otherwise, it is a function of three parameters: 1) the previous frequency that the CR will switch from (f_{i-1}), 2) the current frequency that the CR switched to (f_i), and 3) a technology factor (α). For example, we adopt a linear switching delay function which can be expressed as $\alpha * |f_i - f_{i-1}|$.

Indeed, α depends on many factors like the energy consumed, the error rate, the SNR, and the technology that is used. According to [45], the switching time required for frequency hopping is primarily determined by the design of the phase locked loop (PLL) used in the frequency synthesizer that generates the channel carrier frequencies. A decrease in switching time also comes at the expense of an increase in power dissipation. Table 4.1 shows switching times and the power consumed for switching. Each of these values is for different frequency steps, e.g., the PLL needs 120 μs for 75 MHz steps [46].

Table 4.1: Relationship between switching time and power consumption

Switching time (μs)	**Power(mW)**
0.009	124
0.15	57.6
20	20
70	11.4
120	4.2

The probability of switching, ($Pr(sw)$), is given in the following equation:

$$Pr(sw) = Pr_{i-1}(\mathcal{H}1) * Pr_{i-1}(\mathcal{H}1|\mathcal{H}1) + Pr_{i-1}(\mathcal{H}0) * Pr_{i-1}(\mathcal{H}1|\mathcal{H}0) \tag{4.14}$$

where $Pr_{i-1}(\mathcal{H}1)$ is the probability that channel $i-1$ is busy, $Pr_{i-1}(\mathcal{H}0)$ is the probability that channel $i-1$ is idle, $Pr_{i-1}(\mathcal{H}1|\mathcal{H}1)$ is the probability that channel $i-1$ is busy and it is detected as busy (true detection), and $Pr_{i-1}(\mathcal{H}1|\mathcal{H}0)$ is the probability that channel $i-1$ is idle but it is detected as busy (false alarm). This yields the probability of switching to sense channel i after concluding that the previous channel ($i-1$) is busy, either correctly or mistakenly.

4.6.3 Solution

Algorithm 4 minimizes total sensing time when the CR node is going to sense K channels. In order to find K that achieves the minimum sensing time, we evaluate it iteratively. In each iteration, K will be incremented by 1 and given this K, Algorithm 4 will be solved for total sensing time and thresholds. We keep on incrementing K from 1 towards M until finding the minimum total sensing time, i.e., it decreases, and then starts increasing. At that point, K will be assumed the optimal value.

Some cases are infeasible. For example, if each channel is idle with probability ($Pr(\mathcal{H}0) = 0.6$), then it is infeasible to find an available channel with probability ($\zeta = 0.9$) by searching only 1 or 2 channels even if the CR node conducted perfect sensing with zero false alarm probability. To exclude the infeasible cases, we do not start from $K = 1$, instead, we start it from a larger value, say J. To find J, we initialize J to 1. After that, we assume that $Pr_i(\mathcal{H}1|\mathcal{H}0) = 0$ (which means perfect sensing), $\forall\ i \in [1, M]$. Then, we start incrementing J until the constraint: $1 - \prod_{i=1}^{J}(Pr_i(\mathcal{H}1)) \geq \zeta$ is satisfied. This yields the required value of J.

This search formulation is indeed convex. Lines 1 and 2 are convex for the same reasons mentioned above about the convexity of lines 1 and 2 in monitoring formulation. To prove the convexity of line 3, in general the product of two convex functions is not convex. However, If f and g are convex, both non decreasing (or non increasing), and positive functions on an interval, then $f * g$ is convex. The proof of this claim follows from Jensen's inequality. Line 3 is a product of K Q-functions. The Q-function is convex for input values greater than 0.5, non-increasing and positive function. Therefore, line 3 is convex. Consequently, the search optimization is convex for $\bar{P}_d \geq 0.5$ and $P_f \leq 0.5$.

To reach the global minimum solution quickly, we use a method to find the initial values of the decision variables. Since in algorithm 4, in each iteration, values for γ_i and N_i, $\forall i \in [1, K]$ are found, initial values for N_i are selected, such as 2000. Then, we find the initial values for γ_i using the following equation which is obtained by inverting Equation (4.4):

$$\gamma_i = \left[\frac{Q^{-1}(P_d(i))}{\sqrt{\frac{N_i}{2*SNR_i+1}}} + SNR_i + 1 \right] * \sigma_v^2 \tag{4.15}$$

Using these initial values, and by adding the constraints ($P_f^i \leq 0.5, \forall i \in [1, K]$), convergence to the optimal solution is achieved quickly. We used the sequential quadratic programming (SQP) algorithm [47] for solving this optimization problem which achieves convergence very fast.

4.6.4 Protocol

Channel search is required in case the CR node wants to find an available channel to transmit on, or after doing the in-band sensing and finding that the PU became active. First, the node should use the approach in the previous section to determine the sequential order of channels to be followed during search given it was using the current channel. Then, Algorithm 4 will be applied. As a result of the optimization, the node determines the best number of channels (K), sensing time and threshold value of each of the K channels.

The parameters to Algorithm 4 are: 1) $Pr_i(\mathcal{H}0)$, and SNR_i $\forall$ $i \in [1, M]$. 2) The order in which channels are searched because searching channels is done sequentially. During search, the CR node will follow the order in the sequence that was found by Algorithm 3.

One important thing to notice is that this algorithm is a probabilistic algorithm. This means that after calculating the recommended number of channels to be sensed (K), threshold values (γ_i, $\forall$ $i \in [1, K]$), and sensing time ($ts(i) = N_i$/sampling frequency, $\forall$ $i \in [1, K]$), the CR node will start sensing the channels. It is expected that the node is going to find an available channel by following the sequence and the recommended values with high probability (ζ). It may find an available channel by sensing a fewer number of channels. Also, it may sense the K channels without finding an available channel. In case it did not find an available channel, the CR node can re-apply the optimization problem on the remaining channels, then continue sensing the new K channels with the sensing time and threshold values returned by solving the optimization problem with the new parameters.

4.7 Results and Analysis

We implemented our optimization formulations using Matlab. Throughout the implementation phase, we used the values in table 4.2, unless it is explicitly stated otherwise. We used sequential quadratic programming (SQP) for solving the optimization formulation because it converges very fast.

Table 4.2: Default parameter values used for obtaining results

Parameter	**Value**
$P(\mathcal{H}0)$	0.6
required detection probability ($\bar{P}_d$)	0.94
primary SNR	-16dB
σ_v	1
ζ	0.95
sampling frequency	6Msps
switching time	$120\mu s$ = 720 samples
ϵ used in the optimization	10^{-6}
number of channels (M)	25

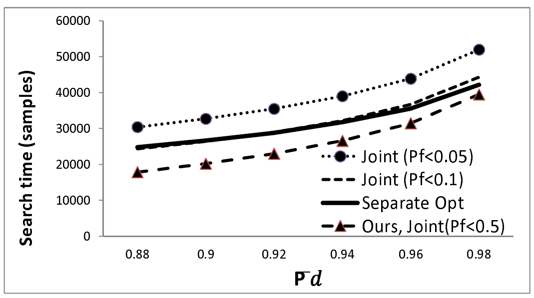
In order to facilitate a fair comparison to other sensing algorithms who try to minimize the false alarm probability to values less than 0.1 or less than 0.05, in our optimization formulations we expanded the acceptable values of false alarm probability. For example, we will compare our algorithm (referred to by curves with $P_f \leq 0.5$) to the approaches that force the false alarm probability to be less than 0.1 and less than 0.05. We will see that relaxing the false alarm probability will reduce required sensing time while protecting the PU.

Non-linear optimization is solved iteratively. It starts from initial values of N_i and γ_i $\forall i \in [1, M]$, then in each iteration, new values for N_i and γ_i are found such that the objective value is closer to the optimal value. This process will be repeated until a stopping criteria is satisfied (difference between the solution of two consecutive iterations is less than ϵ).

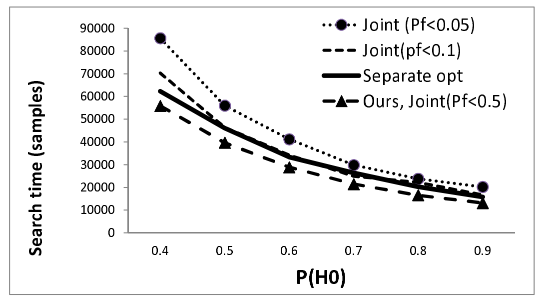
4.7.1 Search Optimization Results

Using the initial values of: $N_i = 5000$, and γ_i according to Equation (4.15) $\forall i \in [1, M]$, we obtained the optimal results for search optimization on average in 25 iterations which means fast convergence.

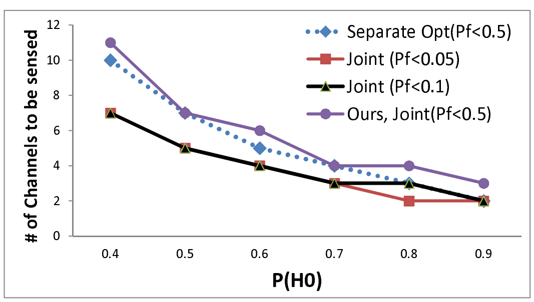
Figures 4.3.a-b show the effects of changing $\bar{P}_d$ and $Pr(\mathcal{H}0)$ on the expected required search time in terms of the total number of samples (N). To calculate the sensing time in seconds, N should be divided by the sampling frequency (6 Msps). From the figures, it is clear that using our approach ($P_f \leq 0.5$) requires shorter sensing time. The curves referred to by "Separate Opt" represent the approach of optimizing the sensing time of each channel separately with false alarm probability ≤ 0.5, and the PU is protected.



(a)



(b)



(c)

Figure 4.3: Effects of: a) $\bar{P}_d$, and b) $P(\mathcal{H}0)$ on the required search time. and c) effect of $P(\mathcal{H}0)$ on number of channels to sense

Figure 4.3.a shows that by increasing the value of $\bar{P}_d$, the required sensing time is increased. This is because larger $\bar{P}_d$ means that the PU tolerates less interference and the results have to be more accurate with higher detection probability. More accurate results can be achieved by sensing for longer time according to Equation (4.4). Figure 4.3.b shows that if the channels have smaller probability of being idle, then the CR needs to do sensing longer to find an

available channel with probability (ζ). This is because it is less probable that the channel is idle, and hence, the CR node has to sense more channels, consequently, longer sensing time. Moreover, our approach requires less search time than optimizing the sensing time of each channel separately.

Figure 4.3.c is related to Figure 4.3.b. The figure shows that sensing more channels with relaxed false alarm probability requires less sensing time than sensing fewer channels with stricter false alarm probability.

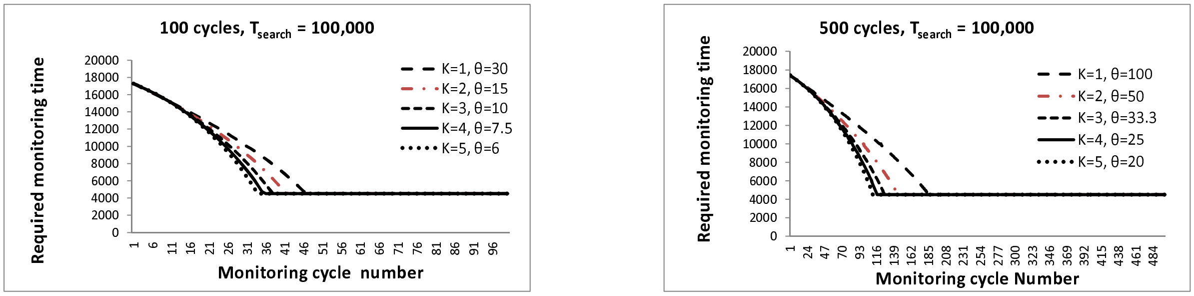
4.7.2 Monitoring Optimization Results

As we mentioned previously, Algorithm 1 is convex. From experiments, and using initial number of samples ($N = 5000$), and using initial threshold value calculated according to Equation (4.15), we obtained the optimal value in 13 iterations on average. In Figure 4.4, we have drawn the results for average search time = 100000 samples. In this section we show the effect on the monitoring time of modeling the PU idle time using different distributions, even if the expected idle period is the same for all distributions. Figure 4.4.a shows the results for a PU model composed of 100 states, while Figure 4.4.b shows the results for a PU model composed of 500 states. The probabilities of the length of the idle period which are shown in Figure 4.1 were therefore found from a Gamma distribution with the parameters K and θ such that the average is fixed. The pdf of the Gamma distribution is given by:

$$f(x; K, \theta) = \frac{1}{\theta^K} \frac{1}{\Gamma(K)} x^{K-1} e^{-\frac{x}{\theta}} \tag{4.16}$$

and its expected value is $K \cdot \theta$. Figure 4.4.a shows the results when the mean is 30, while Figure 4.4.b shows the results when the mean is 100.

Figure 4.4 shows the optimal monitoring time based on in which monitoring cycle the CR node is in. As the monitoring cycle number increases, the probability of being idle decreases, and hence, the monitoring time decreases. This means if the PU is modeled with 100 monitoring cycles or states, the monitoring time in the first monitoring cycle is more than the monitoring time in the 50^{th} monitoring cycle because the probability of the channel being idle in the 50^{th}



(a) PU model of 100 Cycles

(b) PU model of 500 Cycles

Figure 4.4: Effect of in which state the CR node is in, on the required monitoring time

monitoring cycle is less than that in the first monitoring cycle.

4.7.3 Search Sequence Results

Here, we will compare our approach of sorting the channels with: 1) search the channels sequentially which does not consider any other properties of the channels like $P(\mathcal{H}0)$, SNR, or required sensing time. In fact, this will be the best in case the switching time is the dominating factor in search time. 2) the approach that sorts the channels according to the $P(\mathcal{H}0)$. This approach will give priority to the channels that are more probable to be idle. For these two sorting approaches as well our sorting approach, to find the search time, we used Algorithm 4 given the sequential order of the channels according to each of these three approaches.

In this section, we are conducting the simulation on 51 channels in the ranges of 470MHz to 770 MHz. Each channel is 6MHz wide. Each channel has: 1) random SNR between -10 dB and -20 dB, 2) random $P(\mathcal{H}0)$ between 0.2 and 0.8, and 3) random required detection probability $(\bar{(P)}_d)$ between 0.92 and 0.99.

We are considering different switching times that can range from $10\mu s/1MHz$ up to $0.1ms/1MHz$. Figure 4.5 compares our approach to the other two approaches. It is clear that our approach is better than the other approaches because our approach considers both the switching time and the probability of being idle.

Sorting according $P(\mathcal{H}0)$ takes the longest time. This is because $P(\mathcal{H}0)$ does not take into

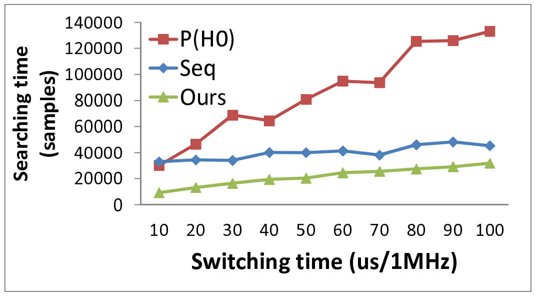


Figure 4.5: Comparison between the three approaches according to search time for different switching times

consideration sensing time. It takes into consideration the probability of being available which is handled by the optimization formulation in Algorithm 4.

4.8 Summary

In this chapter we developed a PU multi-idle states model which allows the CR node to benefit from the last monitoring measurements to calculate the monitoring time for the current monitoring cycle. Based on this model, convex non-linear optimization formulation was introduced for monitoring. Monitoring optimization finds the sensing time, the detection threshold, and the false alarm probability of the channel being used. Search optimization formulation was also introduced. The formulation has more degrees of freedom than previous work, it jointly finds: the sensing time of each channel, the energy detection threshold of each channel, the number of channels to sense, and the false alarm probability of each channel, such that the total search time is minimized, the PUs are protected, and the CR node finds an idle channel with very high probability. The optimization considered channels with different characteristics. Moreover, a heuristic algorithm that sorts the channels in a way to minimize the search time was introduced.

CHAPTER 5 Spectrum Decision for Efficient Routing in Cognitive Radio Networks

5.1 Overview

The cognitive radio (CR) nodes in a cognitive radio network (CRN) do not have license to use specific spectrum band. Instead, they use the spectrum bands of the licensed primary users (PU) without interfering with the PU. When the PU becomes active, interfering CRs should leave to another available spectrum band within the PU's tolerable interference delay (TID). Therefore, CRN operates over wide spectrum bands which span many channels. Since each channel is typically licensed to one PU, this requires that channels be sensed separately. This adds monitoring overhead, where the CR should monitor (sense) the channel every TID, which reduces the throughput. For this reason, the node cannot monitor the whole set of channels.

Deciding which set of channels to monitor affect other functions in the CRN like routing. Work done on routing in literature assumes that each node maintains a set of available channels which is obtained by sensing. Route setup decision will be made based on the available sets at all nodes. However, there may be some other available channels that the node is not aware of their availability which may enhance the routing quality metric. Also, taking into consideration only the sets of channels available at the CR nodes may preclude finding an end-to-end path.

In this chapter, we propose a spectrum decision framework that is complementary to the existing routing protocols. This framework is based on two objectives: 1) enhancing the route quality by sensing a few more channels at some nodes. These channels can enhance the quality by: reducing the switching time, requiring shorter sensing time, or expected to be available for longer time; 2) increasing the probability of finding a path by sensing more channels at some nodes in case the routing protocol did not find a path.

Simulation results show that the proposed framework can result in enhancement that can be as high as 100% over the routing protocols that build their decisions based on the available channels at each node only.

5.2 system Model

The main objective behind this work is to design a spectrum decision framework that will not only consider the set of available channels at each CR node, but also the other channels that are not maintained by the CR nodes and may be available. Our objective is to use this framework to enhance the performance of existing routing protocols, and not to design a new routing protocol that jointly finds the path and the channel to be used on each hop of the path.

We assume that each channel is assigned to one PU who has an exclusive right to use it whenever he wants. If the PU can tolerate interference up to 1 second, then the CR should sense (monitor) the channel periodically every second. If the CR node is maintaining a set of channels, the CR node should sense each of these channels periodically. In addition to the sensing time, the CR node takes some time to switch from one channel to the other. Switching time depends on the frequency step, e.g., to switch from a channel on central frequency, $f1$ MHz, to a channel on central frequency, $f2$ MHz, the switching time will typically be $\alpha * |f1 - f2|$ [31], where α is the switching time per 1 MHz step, and is technology dependent.

We also assume that there exists a routing algorithm that finds the path and the channel to be used on each hop. Therefore, the inputs to our framework are: 1) CRN topology, which consists of the CR nodes and their locations, 2) The outputs of the routing algorithm which are a path from a source to a destination, and the channels selected on each hop, 3) The set of all the channels that the CRN can potentially use, and 4) Some statistics about the PUs activity like the expected active and inactive times, its location, required periodic monitoring time on each CR node, maximum tolerable interference delay which determines how often the CRs should sense the channel.

The output of this framework will be in the form of a set of recommendations to some CRs to sense some channels in order to enhance the routing quality. The recommendations stem from the question: given the output of the routing algorithm which is a path and the channels on that path that are supposed to optimize a specific quality metric, can we enhance the quality of that path by finding other channels on one or more of the hops? The enhancements could be because of finding another channel that requires less monitoring time, less switching time, less access delay, or is being shared between fewer nodes. If the answer to the previous question was yes, then the CR will compare the expected extra cost with the expected benefits that can be gained. If the benefit exceeds the cost, the CR will sense the channel. If the channel found to be available, the CR will start using it. In this chapter, we considered the throughput as the quality metric. However, the same approach can be applied on any routing quality metric.

We assume that all channels have the same bandwidth. We also assume that the activities of the PU on channel k can be represented by a birth/death process as in Figure 5.1, with birth rate (becoming busy), β, and death rate (becoming idle), λ, then the expected time for channel k to be idle within a cycle of activity is ($E_k(\mathcal{H}0) = \frac{1}{\beta}$). Moreover, probability for the PU's channel to be available, $Pr(\mathcal{H}0) = \frac{\lambda}{\lambda+\beta}$, and probability to be busy, $Pr(\mathcal{H}1) = \frac{\beta}{\lambda+\beta}$.

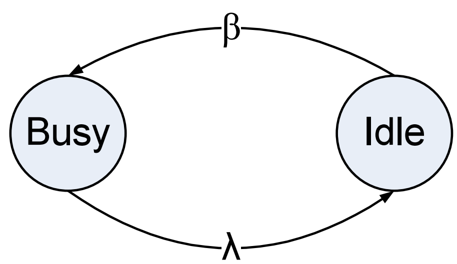


Figure 5.1: PU activity model

5.3 Enhancing Throughput

In this section, we assume that there exists a routing algorithm that finds a path which maximizes the throughput given the sets of sensed channels in the CRN, and the switching times. Therefore, the output of this routing algorithm which forms the input to our framework is a multi-hop path with the channel to be used on each of these hops. Suppose that the cycle length is t_c ms, switching time is α ms/1 MHz, and the channel sensing time for channel i by node j is $ST_j(i)$ ms.

We normalize all of our calculations to the cycle length. Since we assume that all channels have the same bandwidth, then the throughput can be measured by the transmission time, which is equal to:

$$\text{Transmission time} = \text{cycle length - cycle wasted time} \tag{5.1}$$

Therefore, throughout this section we enhance the throughput by increasing the transmission time per cycle. The wasted time is the time due to channel sensing, switching between channels, switching between multi routes, and due to access and sharing a channel.

If a path is composed of multiple intermediate nodes, node (n_i) has the highest wasted time $= t_{wi}$ ms/cycle, and (n_j) has the next highest wasted time $= t_{wj}$ ms/cycle. Then, during each cycle, the destination will not receive for more than $t_c - t_{wi}$ ms. Therefore, if we decide to maximize the throughput, we should find a way to reduce the wasted time at node i. Also, the upper limit of enhancement is to reduce the wasted time at n_i down to (t_{wj}). Therefore, if by finding another channel on node i which reduces the wasted time to $(\hat{t_{wi}} < t_{wj})$, the benefit will be upper limited by t_{wj}, which means the benefit will be $t_{wi} - t_{wj}$. Whereas if $\hat{t_{wi}} > t_{wj}$, the benefit will be $t_{wi} - \hat{t_{wi}}$.

We will discuss the enhancements from applying the proposed framework to multiple cases (Figure 5.2). These cases are not exhaustive, but many other cases can be only simple extension to these. Throughout all the following cases, we will explain the bottleneck with respect to node c, such that it encounters the maximum delay which reduces the transmission time left. Note that neither node d needs be the destination nor node a needs be the source. We are showing only part of the path. Throughout all the cases, we are using some dummy numbers just for the purpose of explanation. The input column shows the result of applying the existing routing protocol which maximizes throughput. The next column shows the result of applying our framework. The last column briefly states the cause of the bottleneck. For example, in the fourth case, using same channel decreases the throughput. Therefore, in the third column, node c chooses to sense another channel and will use it if it is available on both nodes c and d.

5.3.1 Single Path Enhancement Examples

In this section, we will discuss the cases when there is only one path that is node and channel disjoint with all other paths.

Case 1: Suppose that the output from the routing algorithm is as shown in Figure 5.2 Case 1 in the input column. Suppose that node d did not sense channel 1 before making the routing decision such that node d built its routing decision based on the list that it was maintaining, and channel 1 was not among that list. According to the figure, nodes a, b, and c are maintaining channel 1. Therefore, they should sense it periodically. Also, nodes c and d are maintaining channel 5 where they sense it periodically. Since node c is maintaining channels 1 and 5 while node d is maintaining channel 5 only, then it should sense these two channels every cycle, and switch between the channels every cycle, as shown in Figure 5.3.

Our framework recommends enhancement to this routing decision by looking at the bottleneck node, which is node c. This is because node d cannot use channel 1 because it is not within its list of available channels. Therefore, node c needs to switch between channels 1 and 5. Node d can use the idle time during node c switching time, to sense channel 1. According to Figure 5.3, the idle time at d for this case can be given by the equation:

$$IdleTime_d = ST_c(1) + SW(1,5) + ST_c(5) - ST_d(5) \tag{5.2}$$

Assuming symmetric switching, i.e., $SW(1,5) = SW(5,1)$, the time overhead for sensing channel 1 by node d is given by the equation:

$$\begin{aligned} SensingOverhead_d &= SW(5,1) + ST_d(1) + SW(1,5) \\ &= 2 * SW(1,5) + ST_d(1) \end{aligned} \tag{5.3}$$

Then, the cost (C_d) that node d pays is 0 if the sensing overhead is less than the idle time. Otherwise, it is given by the equation:

$$\begin{aligned} C_d &= SensingOverhead_d - IdleTime_d \\ &= SW(1,5) + ST_d(1) - ST_c(1) - ST_c(5) + ST_d(5) \end{aligned} \tag{5.4}$$

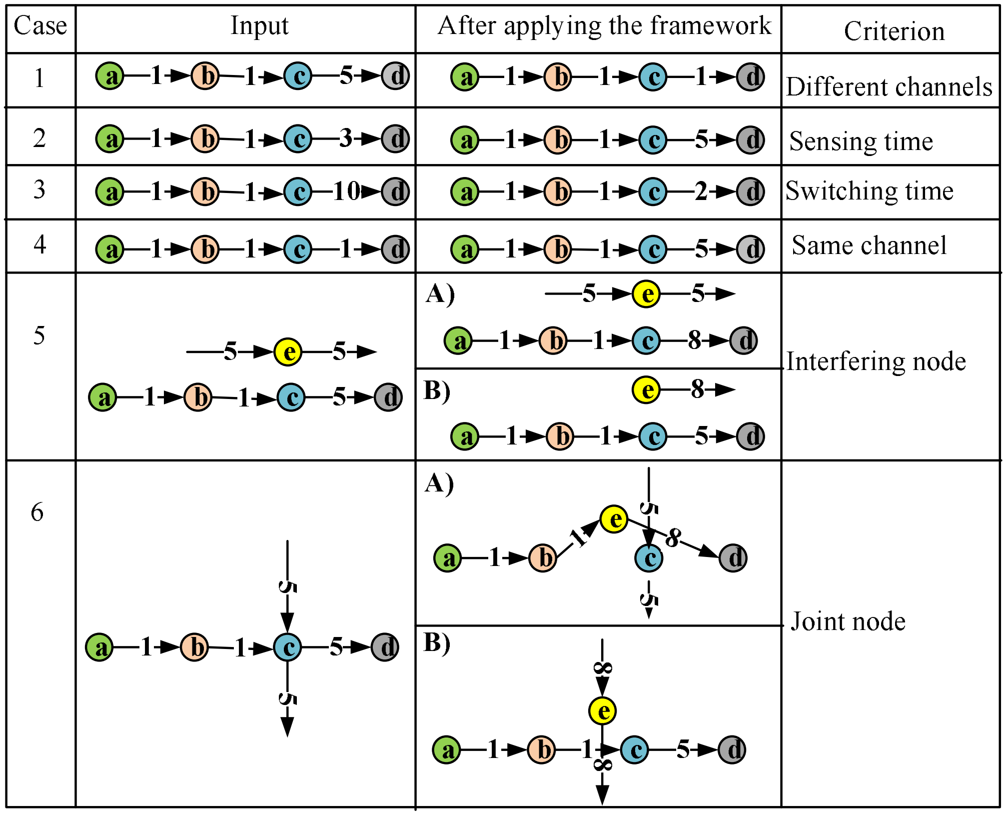


Figure 5.2: Examples on routing enhancements using the proposed framework

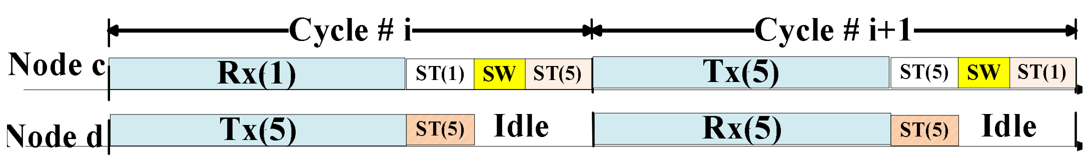


Figure 5.3: Time line for nodes c and d; SW=switching time, ST=sensing time

Then, the expected cost ($\bar{C}_1$) is:

$$\bar{C}_1 = \begin{cases} zero & \text{if } IdleTime_d \geq SensingOverhead_d \\ C_d & \text{otherwise} \end{cases} \tag{5.5}$$

where C_d is given in equation (5.4). On the other hand, the gain per cycle (G) that can be achieved by adding channel 1 to the list of channels maintained by node d is given by the following equation:

$$\begin{aligned} G &= \text{Old wasted time - new wasted time} \\ &= ST_c(1) + SW(1,5) + ST_c(5) - max\{ST_c(1), ST_d(1)\} \end{aligned} \tag{5.6}$$

Suppose that the probability of channel 1 being available at node d is ($Pr_d^1(\mathcal{H}0)$) and being busy is ($Pr_d^1(\mathcal{H}1)$). Then, the expected gain ($\bar{G}_1$) from sensing channel 1 at node d is given by the following:

$$\begin{aligned} \bar{G}_1 &= \text{(gain per cycle) * (expected \#of idle cycles) * Pr(idle)} \\ &= G * \frac{E_1(\mathcal{H}0)}{t_c} * Pr_d^1(\mathcal{H}0) \end{aligned} \tag{5.7}$$

In case node d has sensed channel 1 and knows for sure that it is busy ($Pr_d^1(\mathcal{H}0) = 0$)), then according to Equation (5.7), the benefit will be zero. Therefore, if the cost is larger than zero, it is useless to sense channel 1 at node d. In other words, by comparing the expected cost (5.5) to the expected gain (5.7), we can estimate whether it is cost effective to sense channel 1 at node d or not. Note that in this case we are assuming that node c can send to node d on channel 1 and at the same time node a can send to node b on channel 1. This can happen by using different codes in code division multiple access techniques, or by controlling the transmission power if it is possible. In Case 4, we will show the scenario when it is not possible to simultaneously use the same channel for communication.

Case 2: This case happens when the sensing time of channel 3 at node c ($ST_c(3)$) takes long time such that $SW(1,3) + ST_c(3) > SW(1,5) + ST_c(5)$. The benefit that could be gained from this case is less than Case 1. This may happen if node c is away from the PU that owns channel 3, or because the SNR to the PU is very low, which requires longer sensing time to achieve the required PU detection probability requirement.

Using derivations similar to those in case 1, we derived the final equations for this case. Due to space limitations and since they are similar to those above, we do not show the derivation steps. In this case, the extra cost at node c is the overhead of switching from channel 3 to channel 5, sensing channel 5, and switching back to channel 3. Therefore, the expected cost $(\bar{C}_2)$ can be given by the following equation:

$$\bar{C}_2 = 2 * SW(3,5) + ST_c(5) \tag{5.8}$$

And the expected gain $(\bar{G}_2)$ is:

$$\begin{aligned}\bar{G}_2 = {} & [ST_c(1) + SW(1,3) + ST_c(3) - \\ & \{ST_c(1) + SW(1,5) + ST_c(5)\}] * \frac{E_5(\mathcal{H}0)}{t_c} * Pr_c^5(\mathcal{H}0) \\ & = [SW(1,3) + ST_c(3) - \{SW(1,5) + ST_c(5)\}] * \\ & \frac{E_5(\mathcal{H}0)}{t_c} * Pr_c^5(\mathcal{H}0)\end{aligned} \tag{5.9}$$

If the expected gain calculated by (5.9) is less than zero, this means that using channel 5 will be more expensive than using channel 3, because the sensing plus switching time is larger for channel 5. Moreover, if it is positive, but smaller than the expected cost calculated by (5.8), then it is not beneficial to sense channel 5. However, if it is positive and greater than the expected cost, then node c can sense channel 5. Node d is required to sense channel 5 also. But, we are assuming without loss of generality that the bottleneck is at node c. Therefore, node d can sense channel 5 while node c is sensing channel 5, which means no extra overhead.

Case 3: This case is beneficial in case the switching time is the dominating factor. e.g., in Figure 5.2 Case 3, since node c is required to switch from channel 1 to channel 10 every cycle. If c can find another channel that minimizes the switching plus sensing time like channel 2 in the figure, this will reduce the wasted time. Since we are assuming linear switching time $(\alpha * |f1 - f2|)$, and since node c is required to switch from channel 1 to channel 10, then there is no extra cost for switching because $SW(1,10) = SW(1,2) + SW(2,10)$, otherwise, we should consider the extra switching time. Hence, the expected cost can be given by the equation:

$$\bar{C}_3 = ST_c(2) \tag{5.10}$$

And the expected gain ($\bar{G}_3$) is:

$$\begin{aligned}\bar{G}_3 &= [ST_c(1) + SW(1,10) + ST_c(10) - \\ &\{ST_c(1) + SW(1,2) + ST_c(2)\}] * \frac{E_2(\mathcal{H}0)}{t_c} * Pr_c^2(\mathcal{H}0) \\ &= [SW(1,10) + ST_c(10) - \{SW(1,2) + ST_c(2)\}] * \\ &\frac{E_2(\mathcal{H}0)}{t_c} * Pr_c^2(\mathcal{H}0)\end{aligned} \tag{5.11}$$

Case 4: Under some cases, there are benefits due to switching to other channels and not using the same channel on multiple consecutive hops even if it is available. For example, in Figure 5.2 Case 4, channel 1 is used for the shown three hops. But, if the used channel-sharing method prevents nodes a and c from simultaneously sending on the same channel because node b will not be able to receive data from a when c is transmitting to d. Figure 5.4 shows the time lines for nodes c and d. The figure shows that one third of the time the node is idle because node d cannot send on channel 1 when c is receiving on the same channel. The sensing in this case can be done during the idle time. Moreover, during the idle time, node c can sense some other channels, e.g., channel 5 in Figure 5.2. And the cost will be zero if $(ST_c(5) + 2 * SW(1,5)) \leq (\frac{1}{3} * t_c - ST_c(1))$, which is most probably the case. Otherwise, the expected cost will be:

$$\bar{C}_4 = ST_c(5) + 2 * SW(1,5) - \{\frac{1}{3} * t_c - ST_c(1)\} \tag{5.12}$$

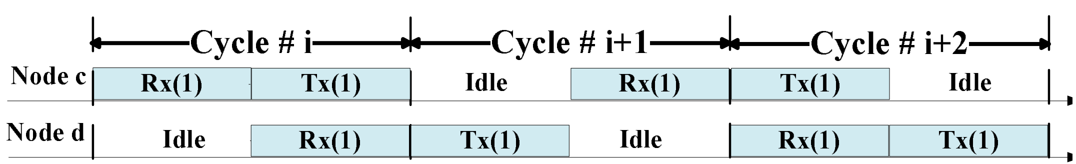


Figure 5.4: Time line for nodes c and d in Case 4

If node c found a channel other than channel 1 to be used between c and d, the time lines for them will be very similar to the time lines in Figure 5.3. Therefore, the expected gain behind using another channel like channel 5 between nodes c and d will be considerable which can be calculated by the following equation:

$$\bar{G}_4 = [\frac{1}{3} * t_c - \{ST_c(1) + SW(1,5) + ST_c(5)\}] * \frac{E_5(\mathcal{H}0)}{t_c} * Pr_d^5(\mathcal{H}0) \tag{5.13}$$

The same method can also be applied between nodes a and b by sensing another channel in order to use it.

5.3.2 Multi-Path Enhancement

In this sub-section, there will be another factor that may affect the throughput of the selected route which is the co-existence of another route that either intersects with the path under study by having a common node, or there is no common node, but there is a channel that is if used on the two routes at the same time, interference happens. Therefore, the two routes alternate on that channel or on that node, which considerably reduces throughput.

Case 5: In this case, as Figure 5.2 Case 5 shows, suppose that there exists a node, e that is an intermediate node on another path, and this node is very close to another node (c). Both nodes use channel 5 for transmission. Since they are close to each other, they cannot transmit at the same time on channel 5 due to interference. Therefore, nodes d and e should alternate on channel 5. The time lines for node c will be very similar to the one in Figure 5.4. Due to space limits, we are not showing it. Therefore, one third of the time, node c is idle, and during this idle time, the node can do the sensing for the channels it maintains. Two possible enhancements are shown in Cases 5.A and 5.B in the output column. We will focus here on Case 5.A because we are studying that input route. Therefore, node c will look for another channel (e.g., channel 8 in Figure 5.2) to be used instead of channel 5. If it is found to be idle, the time line for node c will look like the one in Figure 5.3.

If the time required to sense channel 8 is less than the idle time, it will sense it and the cost will be zero. Otherwise, the cost will be equal to:

$$\begin{aligned} \bar{C}_5 = 2 * SW(5,8) + ST(8) - \{\frac{1}{3} * t_c - \\ (ST_c(1) + SW(1,5) + ST_c(5))\} \end{aligned} \tag{5.14}$$

While the gain is given in the following equation:

$$\bar{G}_5 = [\frac{1}{3} * t_c - \{ST_c(1) + SW(1,8) + ST_c(8)\}] * \frac{E_8(\mathcal{H}0)}{t_c} * Pr_c^8(\mathcal{H}0) \tag{5.15}$$

Note that we did not subtract the sensing time before the enhancement from the first part because node c was idle for $\frac{1}{3}$ of the cycle where it can sense the channels any time within that time.

Case 6: In this case, node c is the bottleneck because it is an intermediate on two routes, which means it will alternate between the two routes, where it will forward the data of the given route half of the time, and forward the data of the other route for the remaining time as Figure 5.5 shows. The dashed rectangles are for the other route. The sw rectangles are switching between channels 1 and 5. The sensing times are one for channel 1 and the other for channel 5. Two enhancements could be done as shown in the two cases 6.A and 6.B in the output column of Figure 5.2. We will explain Case 6.A. Case 6.B will be exactly the same. The main enhancement in Case 6.A is by finding another intermediate node other than node c to forward the data on one of the two routes. This case is different from the previous cases in that it includes finding another node, not just finding another channel.

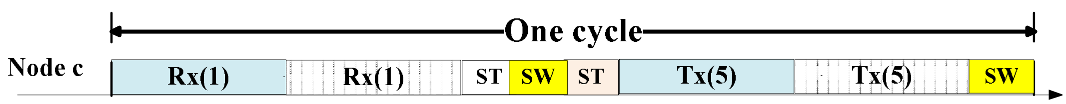


Figure 5.5: Time line for node c in Case 6

This could be initiated by node c sending a message to its neighbors telling them if any of them is physically reachable by nodes b and d, even if there is no common channel known to be available. Suppose node e was found with channel 1 available. Then, we can ask nodes e and d to find a common channel to be used for routing data in this route. Suppose that they are interested in channel 8. Then, the cost will be zero if $[2 * SW(5,8) + ST_d(8)] \leq 0.5 * (t_c + \{ST_c(1) + SW(1,5) + ST_c(5)\})$. Otherwise, the cost will be given by the following equation:

$$\bar{C}_6 = \{2 * SW(5,8) + ST_d(8)\} - \\ \{0.5 * (t_c + \{ST_c(1) + SW(1,5) + ST_c(5)\})\} \tag{5.16}$$

And the expected gain from this enhancement is:

$$\bar{G}_6 = 0.5 * (t_c + ST_c(1) + 2 * sw(1,5) + ST_c(5)) \\ - (ST_c(1) + 2 * sw(1,8) + ST_c(8)) \\ * \frac{E_8(\mathcal{H}0)}{t_c} * Pr_e^8(\mathcal{H}0) \tag{5.17}$$

5.3.3 Protocol

In this sub section, we will introduce a protocol for determining how many channels to sense, which channels to be selected for sensing and when to do the sensing. Case 1 is straight forward, since node d knows that the enhancement can be achieved by looking for the availability of channel 1. Therefore, node d can start sensing channel 1 during the idle time if the expected gain is larger than the expected cost. To find the expected cost and gain, we use the previous derived equation in the previous section.

Regarding the other cases, there are many options for node c to choose from. Therefore, it needs to know what the best channel is to start with. Algorithm 5 shows the general scenario to follow. The idea will be done by sorting the channels descending according to the (payoff = Expected gain - Expected cost).

To find the i^{th} channel to sense, we want to find the channel with the highest payoff among the remaining $(M-i-1)$ channels. The node loops over all the potential $(M-i-1)$ channels. In iteration (j) of the loop, the node first, calculates the gain from using that channel (line 5). Second, it will calculate the cost of inspecting that channel which is the cost of inspecting all the previous $(i-1)$ channels (line 15) plus inspecting the (i^{th}) channel. The cost of inspecting the (i^{th}) channel includes the switching time from the previous channel $(f0)$ to the iterated (j^{th}) channel (line 6). Initially, the current channel is the channel that node c is using for transmission. The first channel to be sensed is the one with the maximum payoff (lines 20-23). Then, the node assumes that the previous channel $(f0)$ is the channel that maximizes the payoff for the current outer loop iteration (Line 23).

The cost will be zero when the idle time is larger than the cost. In cases 2 and 3, the idle time will most probably be smaller than that in Cases 4, 5, and 6. If the maximum payoff was not negative, the node will subtract the cost (which is the sensing plus switching times) from the idle time (line 24).

This procedure will be repeated until the node will not be able to find a channel with positive payoff (lines 17-19). Then, number of channels to sense is known ($NumberOfChToSense$) and the order of the channels the node should follow during sensing is also known (f). If the maximum payoff is negative, the node should not sense any channel ($NumberOfChToSense = 0$), which means there is no possible enhancement to the current situation.

Algorithm 5 : Protocol

```
1: NumberOfChToSense ← 0
2: for i=1:M do
3:    MaxPayoff ← -1 , MaxIndex ← -1
4:    for j=i:M do
5:       G ← Gain(j)
6:       C ← TotalCost(i − 1) + Cost(j, f0)
7:       P ← G − C //P is the payoff which is the objective
8:       if (P ≥ MaxPayoff) then
9:          MaxPayoff ← P
10:         MaxIndex ← j
11:      end if
12:   end for
13:   if (MaxPayoff ≥ 0) then
14:      NumberOfChToSense + +
15:      TotalCost(i) ← TotalCost(i − 1) + Cost(f(MaxIndex), f0)
16:      TotalGain(i) ← Gain(i)
17:   else
18:      return NumberOfChToSense, f //these are the outputs
19:   end if
20:   Temp ← f(x)
21:   f(x) ← f(MaxIndex)
22:   f(MaxIndex) ← Temp
23:   f0 ← f(MaxIndex)
24:   IdleTime ← IdleTime − C
25: end for
```

5.4 Enhance Routing Setup

Since connectivity in CRN is weaker than other networks and frequently changes because it depends on PU behavior, sometimes if a source (s) wants to setup a path to a destination (d), there will be no path. The reason is that on one or more of the hops, there is no common available channel at both nodes at the two ends of that hop. Nevertheless, there may exist a channel that is available, but the nodes are not aware of its availability because they did not sense it. Existing routing protocols will not be able to find the path. However, it is not reasonable to sense all the channels on all the nodes each time there is a need for a route setup, or there is a discontinuity due to PU reappearance. Therefore, we want to know which nodes are better to sense which channels, and when.

Again, we are not designing a new routing protocol. In case the used routing protocol did not find a path from the source to the destination, we will use our framework to find a path which is supposed to increase the probability of finding a path by increasing the number of channels to be checked at some of the nodes.

A common control channel (CCC) is used to flood the route request packet (RRQP) from the source to the destination. Each intermediate node modifies the value of the quality metric (which is defined below). If the intermediate node does not share an available channel with the upstream node, it only increments the number of discontinuities by one and forwards the RRQP to its neighbors through the CCC again.

The destination will receive multiple RRQPs, each contains a two dimensional metric: 1) the quality value and 2) the number of discontinuities value. After that, and depending on the target, the destination node decides to choose: 1) the path with minimum additional setup time (could be the one with the minimum number of discontinuities), 2) the best quality metric value, or 3) a path that achieves best quality metric value such that the number of discontinuities is less than a given constant. In this chapter we will handle the first one. The other two types may be done as a future work.

The quality metric value could be the end-to-end delay, where each intermediate node decides the value of the delay the packet will encounter at the node, add it to the end-to-end delay value in the RRQP, and rebroadcast the RRQP. The node will not re-broadcast the same RRQP again to prevent cycles, except if it has better quality value and/or less number of discontinuities. If the quality metric is throughput, then each node can decide whether it is the bottleneck node or not. In case it is the bottleneck node, then it will modify the RRQP quality value to its throughput. Otherwise, it will not modify it. Then, re-broadcast it.

To estimate the time to find a common channel on one of the hops between two nodes, say

x and y, each node initially has a set of available channels. For channel i, that is within the set of available channels at x, but not within the set of available channels at y, the probability to be available at both x and y will be $1 - Pr(\text{not available at y given it is available at } x) = 1 - Pr(\mathcal{H}1 \text{ at } y | \mathcal{H}0 \text{ at } x)$. Same thing for any channel within the set of available channels at y, but is not within the set of available channels at x, the probability to be available at x and $y = 1 - Pr(\mathcal{H}1 \text{ at } x | \mathcal{H}0 \text{ at } y)$.

On the other hand, for any other channel that is not in the available set of channels neither at x, nor at y, the probability to be available at x and y equals:

$$\begin{aligned} Pr(\mathcal{H}0 \text{ at } x \text{ \& } \mathcal{H}0 \text{ at } y) &= Pr(\mathcal{H}0 \text{ at } x | \mathcal{H}0 \text{ at } y) * Pr(\mathcal{H}0 \text{ at } y) \\ &= Pr(\mathcal{H}0 \text{ at } y | \mathcal{H}0 \text{ at } x) * Pr(\mathcal{H}0 \text{ at } x) \end{aligned} \tag{5.18}$$

The conditional probabilities: $Pr(\mathcal{H}1 \text{ at } x | \mathcal{H}0 \text{ at } y)$, $Pr(\mathcal{H}1 \text{ at } y | \mathcal{H}0 \text{ at } x)$, $Pr(\mathcal{H}0 \text{ at } x | \mathcal{H}0 \text{ at } y)$, and $Pr(\mathcal{H}0 \text{ at } y \text{ and } \mathcal{H}0 \text{ at } y)$, can be calculated from the channel model. For example, [48] models the power received by a CR node by a log-normal random variable. Another approach that can be used is the spectrum cartography maps [49].

If we have multiple discontinuities on one path, then the extra time needed to set up the end-to-end path equals the time consumed at one of the intermediate nodes such that it needs the longest time to find a common available channel with the upstream node and/or the downstream node.

To estimate the minimum time required to find a channel to be available on one hop between two nodes (say x and y), we can follow a way similar to the one in Algorithm 5, but taking into consideration only the cost. We find the cost which is the channel's sensing time plus the switching time from the current channel, multiply the cost by (1- probability of channel to be available at both nodes) to find the expected cost. In each round of the outer loop, we find a channel with the minimum expected cost, and assume it as the current channel, and find the next channel and so on.

To estimate the time, we now know: 1) the order of channels to be followed during search from the previous step, 2) the probability for the channel to be available and 3) the sensing time for each channel. We can calculate the time until finding an available channel with high probability (e.g., ≥ 0.95) which is a geometric distribution.

Each node may find a different sequential order. But, the two nodes should follow the same sequential order during sensing. The two nodes will exchange their sensing decisions on the CCC such that if node x finished sensing channel i first, and found it to be busy, it tells y that it is busy and do not continue sensing. In this case the search time will be minimized because we are taking the minimum sensing time at each node plus some extra communication time overhead.

5.5 Simulation Results

We conducted our simulation using Matlab. In the simulation, we studied how the throughput will be affected by the sensing time and the switching time. Switching time is represented by the switching factor (α) which is the time in ms required per 1 MHz frequency step. When a channel in use becomes busy, the CR searches for another available channel. We did not consider the search time because it is out of the scope of this chapter and it will not affect the results. For the six cases in Figure 5.2, we will show the improvement percentage over the traditional protocols, which refers to the protocols that do not consider sensing other channels. For example, in traditional protocols, if a node maintains a set of 4 channels, then during route setup, the route decision at that node will be made based on these four channels without sensing more channels. However, in our framework, some nodes will consider sensing some other channels that are not within their sets of available channels.

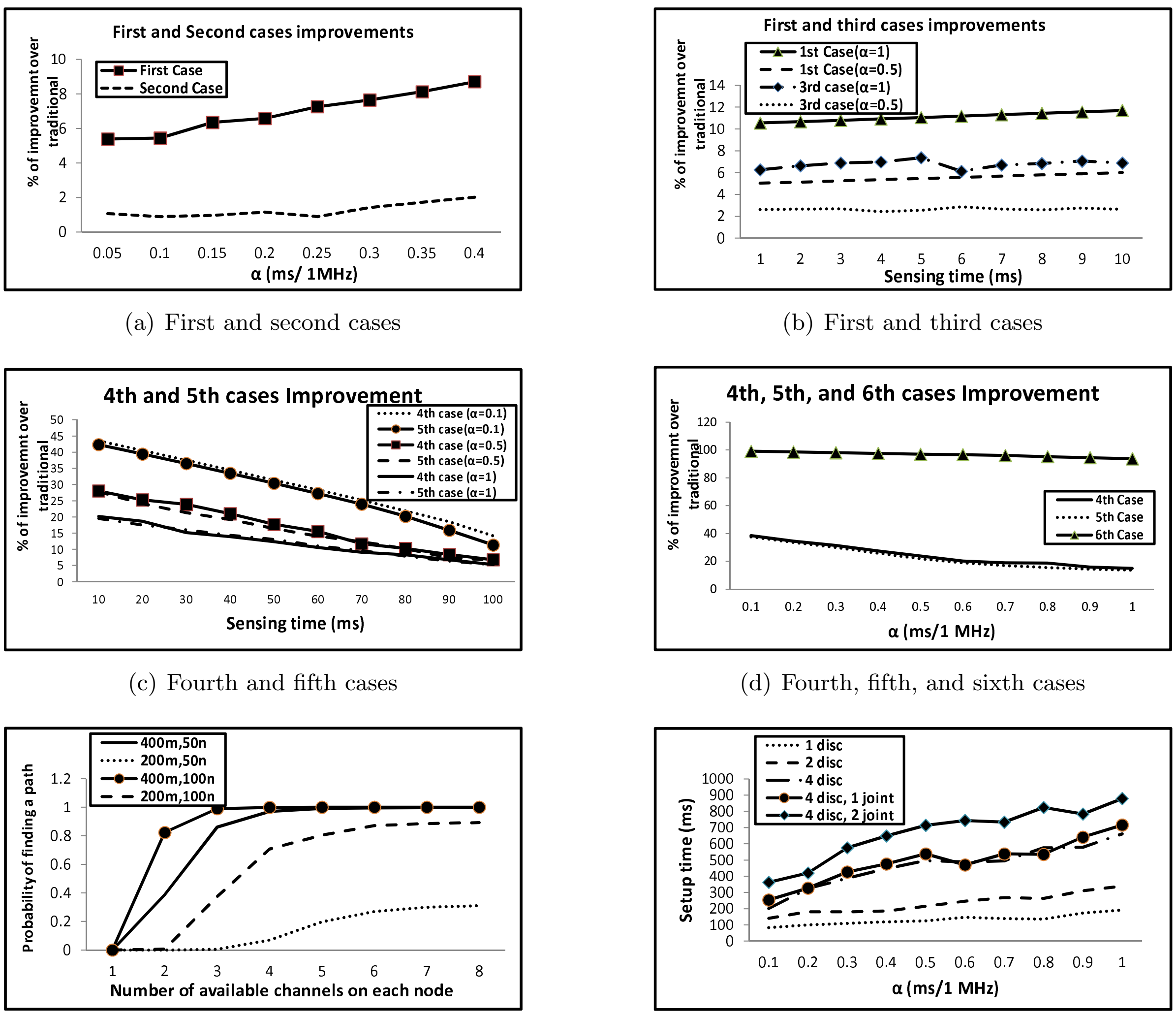
Throughout the simulation, we assumed that the potential number of channels that the CR can work on is 100 channels, and we simulated the operation for 1000 seconds. The cycle length is taken as 1 second, the CR node should sense each channel it maintains every cycle. Channel bandwidth is 6 MHz. λ and β in Figure 5.1 for each PU are selected randomly between 0.01 and 0.1. In the first five cases, we only considered the node that has the bottleneck and the downstream node which are similar to nodes c and d in Figure 5.2, respectively. Since we are studying the throughput, they are enough if we assumed that the bottleneck is at node c. In the sixth case, we considered the three nodes b, c, and d.

To measure the improvement of Case 1, initially, node c will receive on any channel found to be available at c, and will send on any channel found to be available at c and d. In the traditional protocols, they will keep sending on that channel until one of the channels becomes busy. However, in our protocol, node d will check the channel that node c receives on: if it is found to be available, nodes c and d will start using that channel for their communication, otherwise, they will keep on using the same channels. When the channel found to be available, nodes c and d will keep communicating on that channel until it becomes busy. At that point, the two nodes will switch to two new random channels out of those available.

Figures 5.6.$a - b$ show the effect of sensing time and α on the percentage of improvement over traditional routing protocols in the first case. As α increases, the improvement increases. This is because in case the channel found to be idle at d, nodes c and d will start using it for their communication, and the switching overhead at node c will be zero. For traditional protocols, the overhead increases and the throughput decreases as α increases. Therefore, the improvement increases. The enhancement also increases with increasing the sensing time. This is because in case the channel that c receives on is found to be available also at d, node c will not need to sense two channels every cycle. Instead, it will sense one channel which reduces sensing time and increases the throughput.

The second case improvement happens when node c switches between two channels, and the node can find another channel that requires a shorter sensing time. On the other hand, case 3 happens when node c switches between two channels and the switching time takes long time. Therefore, for Case 2, we selected the sensing time for each channel randomly between 1 and 100 ms for nodes c and d, which is a large range in order for some channels to have longer sensing time than others. For case 3, we considered small sensing times compared to switching time. As shown in Figure 5.6.b, the improvement is more in case the switching time is increased because our framework will try to find a channel that reduces switching time. For these two cases, searching time that is needed in our framework to find a better channel is considered as cost, and it is included in the results. When the channel that either node c receives on or it uses to send to d becomes busy, they will switch to a channel randomly. After that in our framework, node c will try to find a channel that reduces the sensing time (Case 2) or a channel that reduces the switching time (Case 3).

Figure 5.6.a shows a small improvement in the second case (up to 2%). This is because the sensing time range is small (between 1 and 100 ms each cycle) compared to the transmission time (1 second cycle).



(a) First and second cases

(b) First and third cases

(c) Fourth and fifth cases

(d) Fourth, fifth, and sixth cases

(e) Probability of finding a path

(f) Route setup time

Figure 5.6: Simulation results

In cases 4 and 5, node c is idle for one third of the time. During this idle time, it can perform sensing. Therefore, in traditional routing protocols, increasing the sensing or switching times will not affect the throughput as long as the sensing plus switching times are less than the idle time (one third of the time). But, in our framework, if the node switched to another channel such that it will not be idle one third of the time, increasing sensing or switching time will reduce the throughput because they reduce the transmission time. This explains why the improvement is decreasing with increasing the switching or the sensing time as Figures 5.6. c and d show. The same thing for the sixth case, but with bigger enhancement. In Case 6, node c is wasting half the time for routing data of the other path. Therefore, if it can find another node to route the data, the throughput will be doubled. For this reason the improvement is close to 100%. In Figure 5.6.d, the sensing time at each node for each channel is selected randomly between 10 and 50 ms.

Enhancing route setup results: To see the effect of our proposed framework on this metric, we deployed 52 and 102 nodes in an area of size 1000m x 1000m. The source node is located at (0,0) and the destination node is located at (1000,1000). The other nodes are at random locations. The total number of channels that the nodes can select from is 20. Each CR node maintains a set of channels which range from 1 up to 8, as Figure 5.6.e shows. These channels are selected by each node randomly out of the 20 channels. To find whether there is a path from the given source to the given destination, we modeled the CRN as a graph. The vertices are the CR nodes, and edge between any two nodes exists if they are within the transmission range of each other and they share a common channel. To find a path, we use the breadth first search approach from the source to the destination.

Figure 5.6.e shows that by increasing the transmission range of the CR node, or by increasing the number of channels the CR node maintains, the probability of finding a path will increase. Also, by increasing the number of nodes in the network, the probability of finding a path increases. In the figure, "400m, n=100" means that the transmission range of the node is 400m and the number of nodes in the network, other than the source and the destination, is 100 node.

Figure 5.6.f shows the expected time required to setup a path. This time is the extra time

required to find common channel between the two nodes that are the two ends of each hop which have no common channel. In all the results, the setup time was less than a second. As the number of discontinuities increases, the setup time increases because we are choosing the maximum time to find a common channel on each hop. We are also showing the setup time for 4 discontinuities with one and two joint discontinuities. We mean by joint is that one of the nodes has no common channel with both upstream and downstream nodes. This does not make a big difference, because in many cases when a node finds a common channel with its upstream node, the same channel will be also available on the downstream node. For this reason, the curve of 4 discontinuities with zero joint and the curve of 4 discontinuities with one joint cross each other multiple times. In this figure, we conducted the simulation on 100 channels. Sensing time of each channel on each node is selected randomly between 1 and 100 ms.

5.6 Summary

In this chapter, we introduced a spectrum decision framework that enhances existing routing protocols. In some cases, the achieved enhancement was as high as 100%. Also, existing routing protocols may not be able to find a path from the source to the destination because there may be no common available channel on one or more hops. The proposed framework was able to find a path with a short extra setup time. The framework concept can be summarized in allowing the CR node to inspect more channels by sensing them. Route quality enhancement stems from finding a channel which requires less sensing time, less switching time, a channel that is shared by less number of nodes, or a channel at one node that will not interfere with other paths. Moreover, another enhancement results when the framework finds another node instead of a node that is intermediate on another path. The framework decides which channels to be sensed, on which nodes, and when it is efficient to sense them, taking into consideration the sensing time, channels switching time, PU expected available time, and the probability of the channel being idle.

CHAPTER 6 A Cross-Layer Routing Protocol (CLRP) in Cognitive Radio Network

6.1 Overview

Routing in cognitive radio networks (CRNs) necessitates a cross-layering approach because routing is based on the information gathered by the sensing which is performed at the physical layer. However, CRN routing protocols proposed in the literature are not truly cross-layer, because the information flow is only from physical layer to network layer, and the monitoring time overhead of the channels, which is required to prevent interference with the PU, is not considered by such protocols. Also, existing routing protocols do not provide the physical layer with information about which channels to sense, which can enhance the routing quality. For example, some channels may be available, and can be used to enhance route quality, but the nodes may not be aware of their availability because they do not sense these channels periodically.

In this work, we introduce a cross-layer routing protocol (CLRP), which considers both the channels that are known to be available at each node, as well as other channels that may be available. The latter channels can be considered using a probabilistic approach. CLRP finds an end to end path, while taking into account the monitoring time, and feeds the physical layer with information about which channels to sense and which nodes should perform the sensing, such that the route quality is enhanced. Using CLRP, we discuss how to enhance the throughput and the stability of the path, and how to increase the probability of finding a path. Simulation results show that CLRP outperforms other cross-layer routing protocols in terms of throughput and stability of the path being setup, and increases the probability of finding an end-to-end path.

6.2 System Model and Problem Definition

The main objective behind this work is to design a CRN routing protocol, that does not only consider the set of available channels at each CR node which are monitored periodically by the CR, but also considers other channels that are not monitored periodically by the CR nodes and may be available. Taking into consideration other channels that may be available with certain probabilities, enhances the performance of existing routing protocols, and increases the probability of finding a path.

We assume that each channel is assigned to one PU who has an exclusive right to use the channel. If the PU can tolerate interference up to 1 second, then the CR should sense (monitor) the channel every second. If the CR node is maintaining a set of channels, the CR node should monitor each of these channels periodically. Also, the CR node spends time to switch from one channel to another, which depends on the frequency step, i.e., to switch from a channel on central frequency, f1 MHz, to a channel on central frequency, f2 MHz, the switching time will typically be $SW(f1, f2) = \alpha * |f1 - f2|$ [31], where α is the switching time per 1 MHz step, and is technology dependent. The monitoring time of each channel is affected by many factors like the signal to noise ratio, required detection probability, received noise, impairments that may affect signal quality like shadowing and fading, and more. Monitoring time is assumed to be different from node to node and from channel to channel.

The inputs to CLRP are: 1) CRN topology, which consists of the CR nodes, their locations, the set of channels known to be available at each node, one source, and one destination, 2) The set of all the channels that the CRN can potentially use, and 3) Statistics about the PUs activity, i.e., the expected active times, their locations, required periodic monitoring time at each CR node, TID which determines how often the CRs should sense the channel.

The output of CLRP will be a path from the source to the destination that is composed of a set of nodes and the channel to be used on each hop. Some of these channels at some relay nodes are available with certain probabilities because the nodes are not sensing them periodically.

We assume that all channels have the same bandwidth. We also assume that the activities of the PU on channel k can be represented by a birth/death process as in Figure 6.1, with birth rate (becoming busy), β, and death rate (becoming idle), λ, then the expected time for channel k to be idle within a cycle of activity is ($E(K) = \frac{1}{\beta}$). Moreover, probability for the PU's channel to be available, $Pr(\mathcal{H}0) = \frac{\lambda}{\lambda+\beta}$, and probability to be busy, $Pr(\mathcal{H}1) = \frac{\beta}{\lambda+\beta}$.

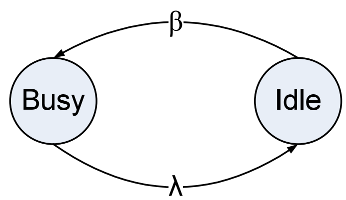


Figure 6.1: PU activity model

The difference that distinguishes the channels that the CR node knows that they are available from the other channels that the CR node is not aware whether they are available or not is the probability to be available ($Pr(\mathcal{H}0)$). For example, for a channel that is within the set of available channels that the CR node senses periodically, $Pr(\mathcal{H}0) = 1$. For a channel that the CR node knows for sure that it is not available, for example it has just sensed it and found it unavailable, $Pr(\mathcal{H}0)$=0. For a channel that is available on one of x's neighbors, i.e., y, then $Pr_x(\mathcal{H}0) = 1$ - Pr(it is not available on x given it is available at y) = 1-$P(\mathcal{H}1$ at $x|\mathcal{H}0$ at y). $Pr(\mathcal{H}0) = \frac{\lambda}{\lambda+\beta}$. The previous conditional probability can be found according to the channel model [48] or using the radio cartography maps [49].

The routing protocol will be initiated by the source node, which floods a route request packet (RRQP) to all of its neighbors. The RRQP contains a table with one entry for each channel. Each entry in the table contains the quality value that will be achieved if the source used that channel for transmission. Each intermediate node modifies the RRQP based on the received RRQPs from its upstream neighbors. When the RRQPs arrive to the destination, it finds which upstream channel and which upstream channel maximize the quality and sends a route reply packet (RRPP) to that node which will be forwarded pack to the source. The RRQP and RRPP will be sent with the help of a common control channel (CCC)

6.3 Enhancing Throughput

Since we are assuming that all the channels have the same bandwidth and same cycle length (T_c), and since each channel must be sensed every cycle, then the throughput can be represented by the transmission time per cycle, i.e., transmission time = cycle length - overhead time per cycle. The overhead time per cycle is the time that the CR node uses for sensing, switching between channels, access the channel, or anything else. In this subsection, we assume that initially each node is subject to a specific load per cycle. For example, node i has a load L_i, where in each cycle the node can at most use $T_c - L_i$ for routing the data on the path under study. The load could be due to sharing the node with other paths, due to sharing channels, due to sensing some other channels, or due to anything else. L_i does not include any overhead from the route being setup. Therefore, if node i is going to use channel x (at central frequency f_x) for reception, and channel y (at f_y) for transmission, the load on node i will become equal to: $L_i + ST_i(x) + ST_i(y) + \alpha * |f_x - f_y|$. Where $ST_i(y)$ is the sensing time of channel y at node i. And the throughput at i if it used channels x and y for reception and transmission, respectively, for the route under study can not exceed $T_c - (L_i + ST_i(x) + ST_i(y) + \alpha * |f_x - f_y|)$.

The process of route setup will be initiated by the source node by building a route request packet (RRQP) to be broadcast to each of its neighbors. The RRQP composed of a table, with each record in the table represents a specific channel. Each record contains two values: the channel ID and the maximum throughput that the source achieves in case it used that channel for transmission. The throughput at the source node for each candidate downstream channel (c), $q_s^d(c) = T_c - L_s - ST_s(c)$. The throughput will be calculated for all candidate channels whether they are known to be available at the source or not. After building the RRQP, the source will broadcast it to its neighbors.

Each intermediate node may receive multiple RRQPs. For example, Algorithm 6 describes how to calculate the best expected upstream quality on each candidate upstream channel,

Algorithm 6 : Finding the expected upstream quality, $q_w^u(c)$ for each channel, c at node w

1: **for** each candidate upstream channel, c **do**
2: $MaxQuality \leftarrow -1$
3: **for** each received RRQP from neighbor, x **do**
4: **if** (c is available at w) **then**
5: $Qua \leftarrow \min\{q_x^d(c), T_c - L_w - ST_w(c)\}$
6: **else**
7: $Qua \leftarrow \min\{q_x(c) * Pr_w^c(\mathcal{H}0), (T_C - L_w - ST_w(c)) * Pr_w^c(\mathcal{H}0)\}$.
8: **end if**
9: **if** ($Qua > MaxQuality$) **then**
10: $MaxQuality \leftarrow Qua$
11: $UpStramNode \leftarrow x$
12: **end if**
13: **end for**
14: $q_w^u(c) \leftarrow MaxQuality$
15: UpStream Node of channel $c \leftarrow UpStramNode$
16: **end for**

when the node w receives multiple RRQPs from its neighbors. The external for loop, loops over all candidate upstream channels, and decides for each of the candidate upstream channels what the best expected upstream quality of that channel is. The internal for loop, loops over the received RRQPs, and decides which upstream neighbor maximizes the expected upstream quality of the channel.

Line 5 means that if channel c is available at w, the expected quality of channel c when w receives from x over the channel c, is the minimum of: 1) the quality value sent from x on channel c, $q_x^d(c)$ and 2) the load on w if it uses channel c for reception. The load equals the cycle time (T_c), minus the initial load on w (L_w), and minus the sensing time of channel c at w, $ST_w(c)$. The minimum is taken because we are studying the throughput which equals the (cycle length - the load) at the node along the path that has the minimum value, hence we are trying to maximize the minimum.

If channel c is available at w with probability $Pr_w^c(\mathcal{H}0)$, line 7 shows the expected upstream quality on channel c. It is similar to line 5, but multiplied by the probability to compute the expected value. This is because w is unsure whether channel c is available or not. Lines 9-12 keep track of the maximum quality (Line 10), and the node that maximizes the quality (Line 11). After the inner for loop finishes, w knows the maximum expected upstream quality that can be achieved if channel c is used for reception, $q_w^u(c)$ (Line 14), and the node that maximizes the upstream quality (Line 15).

Then, w will decide for each candidate downstream channel c, that it can potentially send on, what is the best expected quality value, $q_w^d(c)$ that can be achieved if w used c for transmission, and on which upstream channel and from which upstream node it is better to receive, if the channel c is used for transmission downstream. Algorithm 7 describes how to calculate this.

Algorithm 7 : Finding the expected downstream quality, $q_w^d(c)$ for each channel, c at node w

1: **for** each candidate downstream channel, c **do**
2: $MaxQuality \leftarrow -1$
3: **for** each candidate upstream channel, c_u **do**
4: **if** $(c_u \neq c)$ **then**
5: $Qua \leftarrow \min\{q_w^u(c_u), T_c - L_w - ST_w(c) - ST_w(c_u) - SW(c, c_u)\}$
6: **else**
7: $Qua \leftarrow \min\{q_w^u(c_u), T_c - L_w - ST_w(c)\}$
8: **end if**
9: **if** $(Qua > MaxQuality)$ **then**
10: $MaxQuality \leftarrow Qua$
11: $UpStramCh \leftarrow c_u$
12: **end if**
13: **end for**
14: **if** $(MaxQuality > q_w^d(c))$ **then**
15: $q_w^d(c) \leftarrow MaxQuality$
16: Upstream Channel of $c \leftarrow UpStramCh$
17: $SendRRQP \leftarrow True$
18: **end if**
19: **end for**

In the outer for loop, w loops over all the candidate downstream channels, and for each candidate downstream channel c, it calculates the quality if w used c for transmission. The inner for loop, loops over all the candidate upstream channels, for each candidate upstream channel c_u, w finds the quality if c_u will be used for reception and c will be used for transmission on the route being setup.

Line 5 shows when the upstream channel, c_u is different from the downstream channel c. The quality equals the minimum of $q_w^u(c_u)$ which was calculated in Algorithm 6, and the maximum throughput that can be achieved at w, if channels c_u and c used for reception and transmission, respectively. The maximum throughput that can be achieved is the cycle length, minus the initial load, minus the sensing times of the two channels, and minus the switching time incurred from switching between the two channels to monitor them and to use them. If $c = c_u$ (Line 7), then w senses one channel and the switching overhead equals zero. Downstream quality was not multiplied by the probability of the channel being idle, because it was considered in the upstream quality, and it will be considered at the downstream node.

If the calculated $MaxQuality$ is greater than the old $q_w^d(c)$ of channel c (Lines 14 -18), then $q_w^d(c)$ is modified to $MaxQuality$ (Line 15), and w keeps track of the upstream channel that w is going to receive on, if w used channel c for transmission (Line 16). Also, it modifies the flag $SendRRQP$ which indicates that w should forward the RRQP to its neighbors because it has enhanced quality on one or more channels.

Each node, after it modified and sent the RRQP, may receive new RRQPs. Some of these newly received RRQPs are from some nodes that have already sent the RRQP to the node previously. Since, these new RRQPs must been received because they include some enhanced quality values on some channels. Therefore, the node recalculates the RRQP given all the received RRQPs. It overwrites each entry that resulted in better quality and it does not change other entries. If one or more entries have been changed, the node will re-broadcast the RRQP to its neighbors.

The process continues until the RRQPs arrive to the destination (dst). The destination applies Algorithm 6 to calculate the $q_{dst}^u(c)$ for each channel c. And it decides which upstream channel, say c_u, maximizes the throughput and from which node, say n_u. Then, the destination sends a route reply packet (RRPP) to n_u that it is expecting to receive on channel c_u. Node n_u knows the best upstream channel from Algorithm 7, and the upstream node from Algorithm 6, if it will send on channel c_u. Therefore, it will tell that upstream node that it is going to receive on that channel by forwarding the RRPP packet to that upstream node. The process will be repeated until the RRPP arrives at the source.

Now the path is setup and each node knows on which channel to receive and on which channel to send. The availability of some of these channels is probabilistic. Therefore, any channel that is supposed to be used for routing at a specific node, if it is not within the node's maintained set of available channels (periodically senses them), the node must sense the channel, and use it if it is found to be available. If it is found to be unavailable, the node senses the next channel that maximizes the throughput. One good thing here is that multiple nodes can do sensing in parallel. Also, the nodes that are required to do sensing are known, where not all CRN's nodes should do sensing, and it is also known which channels should be sensed. In our previous work [50], we empirically showed that this additional time takes usually less than a second.

6.4 Enhancing Stability

We define stability as the duration that the path is expected to stay available without interruption by the PUs. One of the differences in routing in CRNs from other types of networks is that routing in CRNs is highly dependent on the PU's behavior, i.e., if the PU became active, then the nodes that are using the PU's channel should leave, which yields disconnected paths. Therefore, some applications may need paths that are expected to stay connected as long as possible regardless of the throughput and regardless of the end-to-end delay.

The stability of a multi-hop path, is measured by the minimum stability on all the hops of the path. For example, if a path is composed of 5 hops and the channels that are used on the five hops are expected to be available for 9, 9, 6, 3, and 10 seconds, respectively, then, the path stability is 3 seconds. The expected available time of a channel can be calculated from the PU behavior as shown in Section 6.2. Therefore, the expected available time of the channel is PU dependent, not CR node dependent. But, the probability of the channel being idle on some nodes will be different among the CR nodes because it depends on the location of the node, and whether the channel is known to be available on one or more of the node's neighbors.

Route setup with enhancing stability quality objective has some similarities to the process of enhancing throughput. However, there are some differences.

1. Line 5 in Algorithm 6, becomes

$$Qua \leftarrow \min\{q_x^d(c), E(c)\} \tag{6.1}$$

   where $E(c)$ is the expected available time of channel c, which is calculated from the PU model as shown in Section 6.2.

2. Line 7 in Algorithm 6, becomes

$$Qua \leftarrow \min\{q_x^d(c) * Pr_w^c(\mathcal{H}0), E(c) * Pr_w^c(\mathcal{H}0)\} \tag{6.2}$$

3. Both Lines 5 and 7 in Algorithm 7, becomes equal to

$$\min\{q_w^u(c), E(c_u)\} \tag{6.3}$$

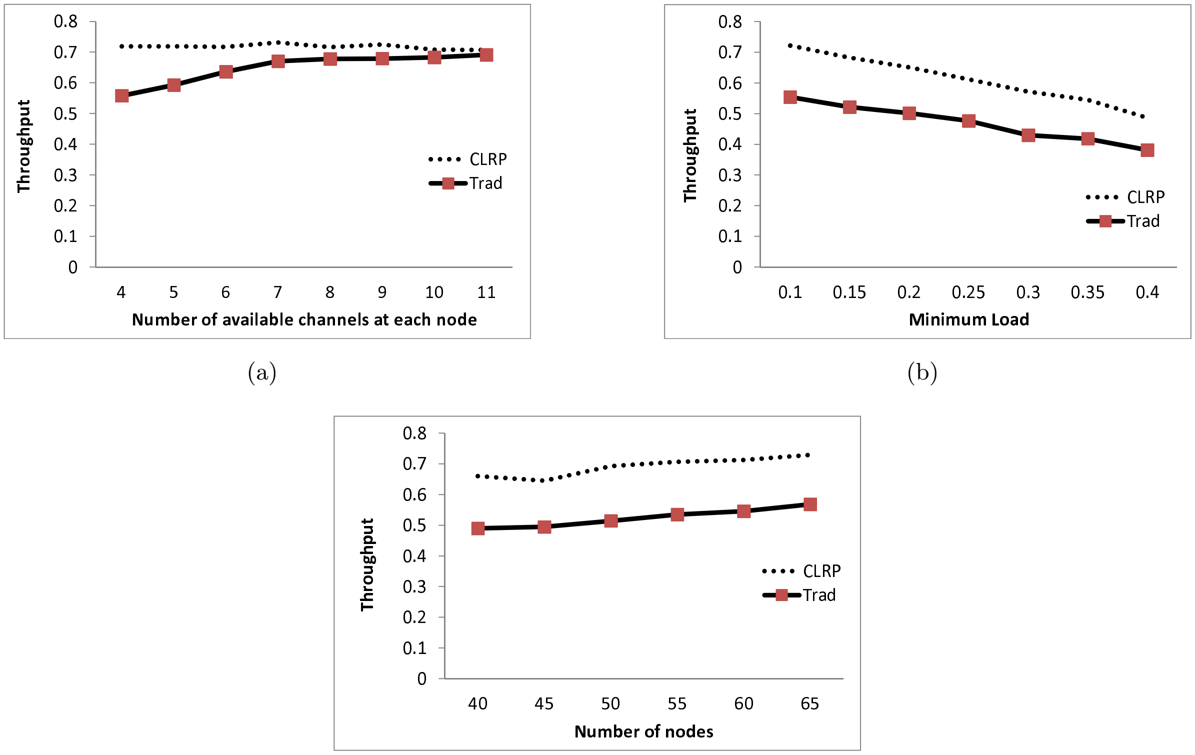
4. To prevent cycles, each node should modify the downstream quality by subtracting a very small number (ϵ) from $q_w^d(c)$ for each channel c, such that always the downstream quality of a channel is less than the upstream quality even for the same channel.

6.5 Simulation Results

We conducted our simulation on Java. We compare our routing approach (CLRP) with the traditional approach (referred to it in the figures by Trad). Traditional approach refers to the protocols that do not consider sensing other channels to make the routing decision. For example, in traditional protocols, if a node maintains a set of 4 channels, where it monitors them periodically, then during route setup, the route decision at that node will be made based on these four channels without sensing extra channels.

Throughout the simulation, we assume the following: total number of candidate channels = 40, PU TID = 1 second, channel bandwidth is 6 MHz, PU are located randomly in a square area between (0,0) and (5000,5000), transmission range of the PU is 2500m, transmission range of the CR is 400 m, λ and β in Figure 6.1 for each PU are selected randomly between 1ms and 100 ms, a CR source is at (0,0), a CR destination is at (1000,1000), 60 other CR nodes are distributed randomly in the square area (0,0) to (1000,1000), load at each CR node is randomly selected between 0.1 and 0.7, Switching α = 1 ms/1MHz, initial number of available channels at each CR node = 4 channels, sensing time of each channel was selected randomly between 1ms and 100ms, and PU status was found randomly based on the probability of being idle or busy. These settings are used during the simulation except stated otherwise.

Figures 6.2.a-c compare the throughput of CLRP with the traditional approach. The throughput in the figures is the achieved throughput after path setup, sensing the channels at the nodes where channels' availabilities are with certain probabilities, and after finding available channels. Each point in these figures is the average of 100 runs. The effect of the



(a) (b) (c)

Figure 6.2: Throughput results. Number of initial available channels in b and c is 4 channels at each node. CR node's transmission range is 400m

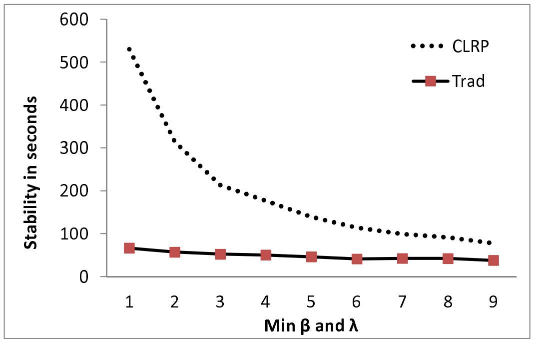
initial number of available channels at each node on the throughput is shown in Figure 6.2.a. The available channels are selected randomly out of the total available channels which are out of the total 40 channels. As the number of available channels increases, the throughput of traditional approaches enhances. This is because, the network will be more connected and the nodes have more options for routing. However, CLRP is not affected by increasing the number of available channels because CLRP considers all the channels, whether they are available or not. The traditional approaches will be close to CLRP as the number of available channels increases. But, this requires too much overhead because the nodes have to do periodic sensing for these channels.

Figure 6.2.b compares CLRP with the traditional approach for minimum load at each node.

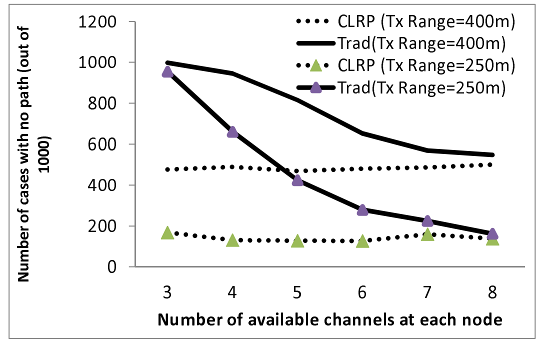
In this figure, the initial load at each node is selected randomly between the minimum load value and 0.7. It is clear that in this case as the minimum load increases, the throughput decreases. During this experiment, 9% of the cases, the traditional approach did not find a path from the source to the destination. However, in CLRP, there is a path in all the simulation runs. Similarly, in Figure 6.2.c, 13.3% of the times there was no path from the source to the destination in the traditional approach. Also, as the number of nodes increases, the throughput gets better because as the network dimensions are fixed, the network gets more connected.

Figure 6.3.a shows the effect of the PU behavior on the stability of the path. In the figure, the values of λ and β for each PU are selected randomly between the value in the figure and 100. Both CLRP and the traditional approach decrease with increasing minimum λ and β. But, CLRP is highly affected with increasing the minimum values because according to the equations in Section 6.2, the expected available time of the channels will be decreased. The decreasing in the traditional approach is slight, because usually there are not many options for the traditional approach, where the path is selected only based on the channels known to be available at each node. Also, the stability equals the minimum stability on all channels along the path.

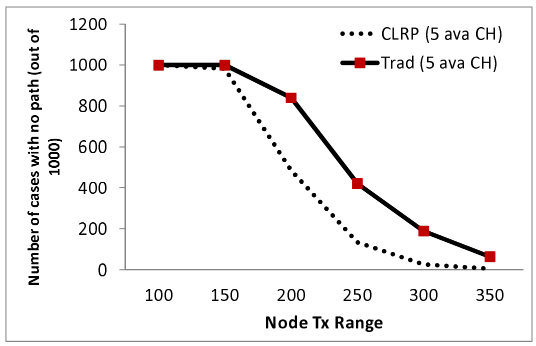
One another benefit of CLRP is increasing the probability of finding a path. For example Figures 6.3.b-c show the effects of changing the number of available channels at each node and the CR transmission range on the number of cases to find a path. Each point is out of 1000 runs. In Figure 6.3.b, the curves when the CR transmission range is 400m were taken on a CR network that spans an area of 2000m x 2000m, while the curves with CR transmission range 250m, the CR network spans an area of 1000m x 1000m. We can see that CLRP is not affected by how many channels are initially available at each node because the CR nodes check all the channels (known to be available or not known). However, CLRP is affected by the CR transmission range because the number of neighbors decreases. On the other hand, the traditional approach is affected by both the CR transmission range and the initial number of available channels at each node.



(a) Stability results



(b)



(c)

Figure 6.3: Figures b and c show number of cases, in which no path was not found from the source to the destination out of 1000 runs

6.6 Summary

In this work we proposed a new approach for routing in cognitive radio networks. The new approach, when finding a route, considers all candidate channels whether they are known to be available at a node, or the node is not aware of their availability because the node is not monitoring these channels periodically. We compared our approach with the traditional approaches which build their route based only on the channels known to be available at each node in the network. Simulation results show that our approach enhances the throughput and the stability of the routes being setup. Also, it increases the probability of finding an end-to-end path.

CHAPTER 7 Conclusions and Future Work

7.1 Conclusions

There are four basic functionalities in cognitive radio network: spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility. These four functions are dependent on each other. However, spectrum sensing assumed to be the key enabling functionality of CRN, because the other three functions are dependent on spectrum sensing. Other functions like routing, accessing the channel, and leaving the channel also depend on spectrum sensing. Therefore, any of the aforementioned functions should consider spectrum sensing.

Fast spectrum sensing increases the quality of service of the CR nodes and increases the utilization of the used spectrum which is the objective behind CR. Also, accurate sensing protects the PU from interference which is a requirement that must be satisfied. However, there is a tradeoff between the speed of sensing and the reliability of sensing which complicates selecting the optimal sensing time that protects the PU.

In this thesis, we studied this tradeoff. In chapter 3 we introduced a framework for cooperative in-band sensing. The target is to allow multiple CR nodes to share the channel such that the sensing efficiency is enhanced and the PU will not suffer from interference for more than the maximum tolerable interference delay (TID) that the PU can tolerate. Usually, the sensing efficiency defined as the ratio of transmission time to the cycle length. A new definition of sensing efficiency was introduced, which is the ratio of the transmitted data size to the summation of the transmitted data size plus the lost data size due to sensing and listening for warning messages. In this framework, the CR node can work in one of two modes, sensing mode and transmission mode. The nodes in sensing mode tells the nodes in transmission mode when the PU becomes active, by sending warning messages. This cooperation was achieved without the need for a common control channel.

In addition, in Chapter 4, we studied this tradeoff for single node sensing, where we designed two optimization formulations. Both of them are non-linear. However, we proved their convexity, and solved them efficiently using algorithms like sequential quadratic programming which converges to the optimal solution quickly. The used underlying sensing method is energy detection. The optimization formulations are for monitoring and searching. The monitoring optimization formulation finds the sensing time, the detection threshold, and the false alarm probability of the channel being monitored. Search optimization formulation has more degrees of freedom than earlier work in the literature, and it jointly finds: the sensing time of each channel, the energy detection threshold of each channel (γ_i), the number of channels to sense, and the false alarm probability of each channel ($P_f(i)$), such that the sensing time is minimized, the PU is protected, and the CR node finds an idle channel with very high probability.

Moreover, we proposed a PU model which models the PU idle state into multi-idle states, each with certain length and certain probability. The model allows the CR node to benefit from its monitoring decisions done in earlier monitoring cycles. Also, we proposed a heuristic approach that sorts the channels in an order that minimizes the expected search time.

In Chapter 5, we proposed a spectrum decision framework that generates recommendations to the physical layer at some CR nodes. The goal behind these recommendations is to enhance the probability of finding an end-to-end path and to enhance the quality of a given route. The proposed idea is complementary to routing protocols, where after the routing protocol finds the path, the proposed spectrum decision solution is applied to enhance the selected path quality. If the existing routing protocol did not find a path, the proposed framework can be applied to find a path. This is because the proposed framework inspects more channels: the ones that are known to be available at the CR nodes, where the CR node monitors them periodically, and the ones that may be available, where the CR node has to check their availability by sensing.

In Chapter 6, we proposed a cross layer routing protocol. The proposed protocol differs from other existing routing protocols in CRNs in: 1) there are two ways of information flow between the network and physical layers: the physical layer tells the network layer which channels are available, and the network layer tells the physical layer of some nodes to sense some extra specific channels. Existing routing algorithms have only information flow from the physical layer to the network layer; 2) Monitoring time is considered; and 3) It considers more channels; channels known to be available and channels that are available with certain probabilities. We used the proposed idea to enhance the throughput and the stability of the path.

7.2 Future Work

In this section, we discuss directions for future work, which can be summarized in:

1. In Chapter 5, we formulated how the proposed framework can enhance the throughput of a given route. We plan to extend this approach to enhance the end-to-end delay and the stability of a given route.

2. In Chapter 5, when the destination receives multiple route request packets, three options can be selected: 1) the path with optimal quality, 2) the path with minimum additional setup time, or 3) a hybrid, i.e., select a path with optimal quality such that number of discontinuities is below a threshold. We selected the minimum setup time in Chapter 5. We plan to investigate the other two options.

3. In Chapter 6, we discussed a true cross layer routing protocol that enhances the throughput and stability. We also plan to study how we can enhance the end-to-end delay.

4. We plan to extend the spectrum decision framework and the cross layer routing protocol such that they consider channels with different capacities and different characteristics.

5. In Chapters 5 and 6, we normalized the throughput to the transmission time per cycle, because we assumed that all the channels have the same bandwidth and same capacity. In case the channels have variable capacities, the transmission time per cycle will not be accurate. Therefore, we plan to study channels with variable capacities and variable data rates.

Bibliography

[1] I.F. Akyildiz, W.Y. Lee, and K.R. Chowdhury. CRAHNs: Cognitive radio ad hoc networks. *Ad Hoc Networks*, 7(5):810–836, 2009.

[2] J. Mitola III and G.Q. Maguire Jr. Cognitive radio: making software radios more personal. *IEEE personal communications*, 6(4):13–18, 1999.

[3] H. Urkowitz. Energy detection of unknown deterministic signals. *Proceedings of the IEEE*, 55(4):523–531, 2005.

[4] D.Y. Seol, H.J. Lim, and G.H. Im. Optimal threshold adaptation with radio environment map for cognitive radio networks. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 2527–2531. IEEE, 2009.

[5] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz. Primary users in cellular networks: A large-scale measurement study. In *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pages 1–11. IEEE, 2008.

[6] Y. Pei, A.T. Hoang, and Y.C. Liang. Sensing-throughput tradeoff in cognitive radio networks: How frequently should spectrum sensing be carried out? In *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007*, pages 1–5, 2007.

[7] A. Ghasemi and E.S. Sousa. Optimization of spectrum sensing for opportunistic spectrum access in cognitive radio networks. In *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC)*, pages 1022–1026, 2007.

[8] P. Wang, L. Xiao, S. Zhou, and J. Wang. Optimization of detection time for channel efficiency in cognitive radio systems. In *IEEE Wireless Communications and Networking Conference, 2007. WCNC 2007*, pages 111–115, 2007.

[9] H. Kim and K.G. Shin. Efficient discovery of spectrum opportunities with MAC-layer sensing in cognitive radio networks. *IEEE Transactions on Mobile Computing*, pages 533–545, 2007.

[10] H. Kim and K.G. Shin. Fast discovery of spectrum opportunities in cognitive radio networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pages 1–12. IEEE, 2008.

[11] D. Cabric, S.M. Mishra, and R.W. Brodersen. Implementation issues in spectrum sensing for cognitive radios. In *Asilomar Conference on Signals, Systems, and Computers*, volume 1, pages 772–776. Citeseer, 2004.

[12] W.Y. Lee and I.F. Akyildiz. Optimal spectrum sensing framework for cognitive radio networks. *IEEE Transactions on Wireless Communications*, 7(10):3845–3857, 2008.

[13] Y.C. Liang, Y. Zeng, E.C.Y. Peh, and A.T. Hoang. Sensing-throughput tradeoff for cognitive radio networks. *IEEE Transactions on Wireless Communications*, 7(4):1326–1337, 2008.

[14] Q. Zhao, S. Geirhofer, L. Tong, and B.M. Sadler. Optimal dynamic spectrum access via periodic channel sensing. In *Proc. Wireless Communications and Networking Conference (WCNC)*. Citeseer, 2007.

[15] M. Cesana, F. Cuomo, and E. Ekici. Routing in cognitive radio networks: Challenges and solutions. *Ad Hoc Networks*, 9(3):228–248, 2011.

[16] ET FCC. Docket no. 08-260,. *Second Report and Order and Memorandum Opinion and Order, Nov*, 2008.

[17] E. Jung and X. Liu. Opportunistic spectrum access in heterogeneous user environments. In *Proceedings of the IEEE DySPAN*, 2008.

[18] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar. IEEE 802.22: an introduction to the first wireless standard based on cognitive radios. *Journal of communications*, 1(1):38–47, 2006.

[19] S.M. Mishra, A. Sahai, and R.W. Brodersen. Cooperative sensing among cognitive radios. In *Proc. of the IEEE International Conference on Communications (ICC)*. Citeseer, 2006.

[20] I.F. Akyildiz, B.F. Lo, and R. Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Communication*, 2010.

[21] A. Ghasemi and E.S. Sousa. Collaborative spectrum sensing for opportunistic access in fading environments. In *Proc. IEEE DySPAN*, volume 2005, pages 131–136, 2005.

[22] L. Lai, H. El Gamal, H. Jiang, and H.V. Poor. Cognitive medium access: Exploration, exploitation and competition. *IEEE/ACM Trans. on Networking*, 2007.

[23] B. Wild and K. Ramchandran. Detecting primary receivers for cognitive radio applications. In *Proc. IEEE DySPAN*, volume 2005, pages 124–130. Citeseer, 2005.

[24] X. Chen, Z. Bie, and W. Wu. Detection efficiency of cooperative spectrum sensing in cognitive radio network. *The Journal of China Universities of Posts and Telecommunications*, 15(3):1–7, 2008.

[25] X. Liu and Z. Ding. ESCAPE: a channel evacuation protocol for spectrum-agile networks. In *Proceedings of the IEEE DySPAN*, 2007.

[26] Z. Quan, S. Cui, A.H. Sayed, and H.V. Poor. Wideband spectrum sensing in cognitive radio networks. In *Communications, 2008. ICC'08. IEEE International Conference on*, pages 901–906. IEEE, 2008.

[27] E. Peh and Y.C. Liang. Optimization for cooperative sensing in cognitive radio networks. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pages 27–32. IEEE, 2007.

[28] I. Pefkianakis, S.H.Y. Wong, and S. Lu. Samer: spectrum aware mesh routing in cognitive radio networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pages 1–5. Ieee, 2008.

[29] A. Sampath, L. Yang, L. Cao, H. Zheng, and B.Y. Zhao. High throughput spectrum-aware routing for cognitive radio networks. *Proc. of IEEE Crowncom*, 2008.

[30] L. Ding, T. Melodia, S. Batalama, and M.J. Medley. Rosa: Distributed joint routing and dynamic spectrum allocation in cognitive radio ad hoc networks. In *Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 13–20. ACM, 2009.

[31] G. Cheng, W. Liu, Y. Li, and W. Cheng. Spectrum aware on-demand routing in cognitive radio networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pages 571–574. IEEE, 2007.

[32] G. Cheng, W. Liu, Y. Li, and W. Cheng. Joint on-demand routing and spectrum assignment in cognitive radio networks. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 6499–6503. Ieee, 2007.

[33] C. Xin, B. Xie, and C.C. Shen. A novel layered graph model for topology formation and routing in dynamic spectrum access networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 308–317. IEEE, 2005.

[34] Y.T. Hou, Y. Shi, and H.D. Sherali. Optimal spectrum sharing for multi-hop software defined radio networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1–9. IEEE, 2007.

[35] M.B.H. Weiss, S. Delaere, and W.H. Lehr. Sensing as a service: An exploration into practical implementations of dsa. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–8. IEEE, 2010.

[36] H. Kim and K.G. Shin. In-band spectrum sensing in cognitive radio networks: energy detection or feature detection? In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 14–25. ACM, 2008.

[37] K. Shin, A. Abraham, and S.Y. Han. Self organizing sensor networks using intelligent clustering. *Lecture Notes in Computer Science*, 3983:40, 2006.

[38] Y. Zhao, B. Le, and J.H. Reed. Network support–The radio environment map. *Cognitive radio technology*, page 337, 2006.

[39] Y. Zhao, J. Gaeddert, K.K. Bae, and J.H. Reed. Radio environment map enabled situation-aware cognitive radio learning algorithms. In *Software Defined Radio Forum (SDRF) technical conference*, 2006.

[40] Y. Zhao, J.H. Reed, S. Mao, and K.K. Bae. Overhead analysis for radio environment mapenabled cognitive radio networks. In *Networking Technologies for Software Defined Radio Networks, 2006. SDR'06.1 st IEEE Workshop on*, pages 18–25. IEEE, 2007.

[41] M. Di Felice, K.R. Chowdhury, and L. Bononi. Analyzing the potential of cooperative cognitive radio technology on inter-vehicle communication. In *Wireless Days (WD), 2010 IFIP*, pages 1–6. IEEE.

[42] C. Cordeiro, K. Challapali, D. Birru, S. Shankar, et al. IEEE 802.22: the first worldwide wireless standard based on cognitive radios. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 328–337. Ieee, 2005.

[43] L. Kleinrock. Queueing systems. volume 1: Theory. 1975.

[44] R. Tandra and A. Sahai. SNR walls for signal detection. *IEEE Journal of Selected Topics in Signal Processing*, 2(1):4–17, 2008.

[45] L. Luo, N.M. Neihart, S. Roy, and D.J. Allstot. A two-stage sensing technique for dynamic spectrum access. *Wireless Communications, IEEE Transactions on*, 8(6):3028–3037, 2009.

[46] S. Shin, K. Lee, and S.M. Kang. 4.2 W CMOS frequency synthesizer for 2.4 GHz ZigBee application with fast settling time performance. In *Microwave Symposium Digest, 2006. IEEE MTT-S International*, pages 411–414. IEEE, 2007.

[47] P.T. Boggs and J.W. Tolle. Sequential quadratic programming. *Acta numerica*, 4:1–51, 1995.

[48] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[49] E. Dall'Anese, S.J. Kim, and G.B. Giannakis. Channel gain map tracking via distributed kriging. *Vehicular Technology, IEEE Transactions on*, 60(3):1205–1211, 2011.

[50] R. Saifan, A.E. Kamal, and Y. Guan. Spectrum decision for efficient routing in cognitive radio network. In *Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 8th International Conference on*. IEEE, 2012.